TOWARD AN EXTRAPOLATION OF THE SIMULATED ANNEALING CONVERGENCE THEORY ONTO THE SIMPLE GENETIC ALGORITHM

By

THOMAS E. DAVIS

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

1991

UNIVERSITY OF FLORIDA
3 1262 08552 3479

ACKNOWLEDGEMENTS

The author is extremely fortunate in having the assistance of several talented academicians during the conduct of the research program reported in this dissertation. Notably, Professor Jose Principe, who supervised this work and contributed several key ideas developed herein, proved a very valuable source of encouragement and support. Also, Professor Murali Rao assisted very substantially in enforcing mathematical rigor, especially in the formulation of the Markov chain appendices. Professors Antonio Arroyo and Donald Childers, who served on the committee overseeing this work, are remembered fondly for the constructive comments they provided during the visits the author made to Gainesville while conducting this research, as well as for productive associations while the author was in residence. Also, the assistance provided by Professor Eugene Chenette from the Eglin Graduate Center in dealing with a variety of administrative complications, as well as his service on the committee overseeing this work, is graciously acknowledged.

Additionally, the generous support of the US Air Force Armament Laboratory to this work is sincerely appreciated. The author's management chain, notably Mr. Lynn Deibler, Lt. Col. Rex Franklin, Dr. Eugene Youngblood and Lt. Col. Tom Callen, provided continual encouragement and working condition flexibility. Without their support, this activity would likely not have been possible.

The Computer Science Directorate at Eglin provided some exceptionally valuable computer support, on the Eglin AFB Cray Y-MP, under very flexible conditions. Some of the insights gained during the conduct of this research would be very difficult, and perhaps impossible, to attain through any method other than simulation. Members of the staff at the Computer Science Directorate who contributed significantly, especially in

helping the VMS-inclined author through the UNIX maze, include Mr. Eddie Blackwell, Mr. Ben McKinnon and Mr. Danny Majors. Mr. Bill Clements, who made the computing resources available, and Mr. Calvin George, who helped the author arrange the support, are also gratefully acknowledged.

Finally, the author wishes to thank Sumiko, who entered his life during the conduct of this research program, for the support and understanding whose need she never fails to anticipate.

TABLE OF CONTENTS

	page
ACKNOWLEDGEMENTS	ii
ABSTRACT	vii
SECTIONS	
1 INTRODUCTION	1
1.1 Non-Convex Combinatorial Optimization and Stochastic Search Algorithms	1
1.2 Organization	2
2 SIMULATED ANNEALING	7
2.1 Overview	7
2.2 Statistical Mechanics and Annealing of Solids	7
2.3 Combinatorial Optimization by Simulated Annealing	9
2.4 Theoretical Foundations of Simulated Annealing	10
3 THE GENETIC ALGORITHM	20
3.1 Overview	20
3.2 The Simple Genetic Algorithm Operators	21
3.3 Building Blocks, Schemata and the Fundamental Theorem	23
3.4 An Assessment of the Genetic Algorithm Theoretical Foundation	26
4 A MARKOV CHAIN MODEL OF THE SIMPLE GENETIC ALGORITHM	28
4.1 Overview	28
4.2 The Markov Chain Model	28
4.3 The State Behavior of the Simple Genetic Algorithm	30
5 SOME EMPIRICAL RESULTS	42
5.1 Overview	42
5.2 State Space Enumeration	43
5.3 Reward Function Data	46
5.4 Conditional Probabilities vs α	48
5.5 Converged Limiting Stationary Distributions	52

page

6 THE CRAMER'S RULE FORMULATION OF THE STATIONARY DISTRIBUTION 66

6.1 Overview 66
6.2 The Stationary Distribution Description 66
6.3 Positivity of the Stationary Distribution Components 71
6.4 The Indeterminate Form at $\alpha = 0$ 72

7 THE ZERO MUTATION PROBABILITY STATIONARY DISTRIBUTION LIMIT 73

7.1 Overview 73
7.2 Functional Form of the Stationary Distribution 73
7.3 The Absorbing State Rows of $|\overline{P} - \overline{I}|$ and $|\overline{P}_{\overline{m}} - \overline{I}|$ 75
7.4 Reformulation of Propositions 6.7 and 6.8 77
7.5 The Stationary Distribution Limit 81

8 A MONOTONIC MUTATION PROBABILITY ERGODICITY BOUND 86

8.1 Overview 86
8.2 A Weak Ergodicity Bound 87
8.3 Strong Ergodicity 88
8.4 Comparison With the Simulated Annealing Parameter Bound 91

9 REPRESENTATION OF THE STATIONARY DISTRIBUTION SOLUTION 92

9.1 Overview 92
9.2 The Limiting Case $\alpha = 1$ 93
9.3 The General Case $0 < \alpha \leq 1$ 97
9.4 The Limiting Case $\alpha \rightarrow 0^{+}$ 109
9.5 Extending the Stationary Distribution Representation 116

10 CONCLUSIONS AND FUTURE DIRECTION 120

10.1 Summary 120
10.2 Contributions of the Research 124
10.3 Future Direction 125

APPENDICES

A DISCRETE TIME FINITE STATE MARKOV CHAINS 126

A.1 Introduction 126
A.2 Elementary Definitions 126
A.3 Time-Homogeneous Markov Chains 128
A.4 Inhomogeneous Markov Chains 130

page

B THE PERRON-FROBENIUS THEOREM AND STOCHASTIC MATRICES 132

B.1 Introduction 132
B.2 The Perron-Frobenius Theorem and Ancillary Results for Primitive Matrices 132
B.3 The Perron-Frobenius Theorem for Stochastic Matrices 134

C VANDERMONDE DETERMINANTS, SYMMETRIC AND ALTERNATING POLYNOMIALS 137

C.1 Introduction 137
C.2 Evaluation of Vandermonde Determinants 138
C.3 Symmetric (and Alternating) Polynomials 139
C.4 Quasi-Symmetric (and Quasi-Alternating) Polynomials 142

D COMPUTER LISTINGS 145

D.1 Introduction 145
D.2 Main Program Listings 145
D.3 Library Listings 152

REFERENCES 163

BIOGRAPHICAL SKETCH 166

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

TOWARD AN EXTRAPOLATION OF THE SIMULATED ANNEALING CONVERGENCE THEORY ONTO THE SIMPLE GENETIC ALGORITHM

By

THOMAS E. DAVIS

May 1991

Chairman: Professor Jose C. Principe
Major Department: Electrical Engineering

Simulated annealing and the genetic algorithm are stochastic relaxation search techniques suitable for application to a wide variety of combinatorial complexity nonconvex optimization problems. Each produces a sequence of candidate solutions (or populations of candidate solutions) to the underlying optimization problem, and the purpose of both algorithms is to generate sequences biased toward solutions which optimize the objective function.

The appeal of simulated annealing is that it provides asymptotic convergence to a globally optimal solution. A substantial body of knowledge exists concerning the algorithm convergence behavior. It is based upon a nonstationary Markov chain algorithm model. No genetic algorithm model comparable in scope exists in the literature. This work constitutes an attempt to provide such a model and accompanying convergence theory by extrapolating the simulated annealing results onto the genetic algorithm. A prerequisite, developed herein, is a nonstationary Markov chain genetic algorithm model.

The essence of the simulated annealing theory is demonstration of (1) existence of a unique asymptotic probability distribution (stationary distribution) for the stationary Markov chain corresponding to every strictly positive constant value of an algorithm control parameter (absolute temperature), (2) existence of a stationary distribution limit as the control parameter approaches zero, (3) the desired behavior of the stationary distribution limit (i.e. optimal solution with probability one) and (4) sufficient conditions on the algorithm control parameter to ensure that the nonstationary algorithm achieves (asymptotically) the limiting distribution. With the exception of (3), this work adapts that methodology to the genetic algorithm Markov chain model employing a genetic operator parameter (mutation probability) as the algorithm control parameter. The results include a mutation probability control parameter bound analogous to (and asymptotically superior to) the conventional simulated annealing parameter bounds, and a framework for representing the genetic algorithm stationary distribution components at all consistent fixed control parameter values, including zero.

The genetic algorithm stationary distribution limit has nonzero components corresponding to all solutions. Thus, the simulated annealing global optimality convergence result does not extrapolate. However, both empirical and theoretical evidence is provided which suggests that the desired limiting behavior can be approached by suitably adjusting the algorithm parameters.

SECTION 1

INTRODUCTION

1.1 Non-Convex Combinatorial Optimization and Stochastic Search Algorithms

A wide variety of engineering applications lend themselves to formulations which require the solution of combinatorial optimization problems. Typically, the optimization problem is nonconvex and is defined over a very high dimensionality search space (e.g. inverse vision problems, in which an image array of 512X512 pixels at 8 bits/pixel might be encountered, resulting in a search space dimensionality of ~2M). Consequently, direct solution is usually intractable.

An alternative to direct solution is to select one of a variety of iterative improvement solution techniques, usually some variant of gradient search. But by definition, deterministic iterative improvement techniques terminate in local extrema, and they ordinarily provide no means of assessing the amount by which the selected local extremum deviates from the global extremum. A typical means of avoiding local extrema entrapment is to implement the iterative improvement solution method stochastically.

The most commonly employed stochastic algorithm approach to combinatorial optimization is simulated annealing [KiGe83, LaAa87], which is also sometimes referred to as probabilistic hill climbing [RoSa85]. It exploits the analogy of combinatorial optimization to the annealing of crystalline solids, in which a solid is cooled very gradually from some elevated temperature and thereby allowed to relax toward its low energy states. The appeal of the algorithm class derives from the fact that provided certain constraints on an algorithm control parameter (analogous to absolute temperature) are observed, asymptotic convergence to a global extremum is guaranteed.

The key limitation of simulated annealing is that the convergence behavior is asymptotic. Thus global optimality is obtained only after an infinite number of algorithm iterations. The rate of convergence to optimality is determined by a nonnegative algorithm control parameter whose ideal value is zero and which must observe a lower bound in order to assure coherent algorithm behavior. The best available known bound for the parameter, the annealing schedule bound, is of the form K/log(k) where k is the iteration index and K is a parameter independent of k [GeGe84, MiRo85].

Another combinatorial optimization stochastic search technique reported in the literature is the genetic algorithm [Davi87, Gold83, Gold89a, Gref85, Gref87]. It emulates the evolution of biological systems by employing a set of stochastic operators (e.g. reproduction, crossover and mutation) to transform a population of candidate solutions to the underlying optimization problem into a new (descendent) population. It has some features which suggest that it may provide significantly improved convergence behavior over simulated annealing on certain types of optimization problems. However, the nature of the genetic operators and their influence on algorithm behavior is only understood in general terms. No complete theoretical model of the algorithm exists in the literature. The fundamental goal of the work reported here is to provide a theoretical framework for analyzing the algorithm based upon the asymptotic probability distribution of the solution sequences which it produces. The work reported herein includes significant progress on the key intermediate steps to achieving that goal.

1.2 Organization

The remaining sections of this paper are organized as follows. Sections 2 and 3 are background reviews of the simulated annealing and genetic algorithm literature respectively. Section 2 places considerable emphasis on the methodology employed to yield the asymptotic convergence results which are the theoretical foundation of the simulated annealing algorithm. That methodology appeals heavily to the theory of inhomogeneous (nonstationary) Markov chains and their asymptotic state probability distributions. The

essence of the simulated annealing convergence theory is a set of sufficient conditions to ensure that the asymptotic probability distribution of the Markov chain which represents the algorithm is independent of its starting state and has probability zero for all states corresponding to sub-optimal solutions.

Section 3 begins with a verbal description of the three fundamental stochastic operators employed in genetic algorithms (i.e. reproduction, crossover and mutation), and proceeds to review the existing theoretical foundation of the algorithm class. A conclusion of that section is that while certain important theoretical results exist, notably the so-called schema theorem and some work on a problem construct referred to as the minimal deceptive problem, the genetic algorithm lacks the theoretical foundation necessary to either compare it with simulated annealing or to answer key questions concerning the design of a genetic algorithm for a given application.

The author's contribution to this work begins with Section 4. The major result of that section is a very general, nonstationary Markov chain model of the variants of the genetic algorithm which employ combinations of the three fundamental genetic algorithm operators. The model is tailored to resemble that employed in developing the simulated annealing methodology, and in that regard, the genetic algorithm mutation operator is shown to provide a function very similar to that of the simulated annealing absolute temperature analog. Specifically, the stationary algorithm corresponding to every constant value of the mutation probability parameter satisfying $0<p_m<1$ possesses a unique asymptotic probability distribution (stationary distribution).

The key results of Section 4 are state transition matrices, formulated in terms of the algorithm parameters and objective function of the underlying optimization problem, for one, two and three-operator variants of the algorithm. Due to the nature of the genetic operators, the state transition matrices exhibit an extremely high degree of symmetry. These matrices, and some key related results, are used extensively in later sections.

Section 5 digresses briefly from the theoretical development to produce and examine some empirical work based upon the algorithm model. The presentation is not, nor is it intended to be, a thorough empirical study. It is provided to help fix some of the algorithm model state space and asymptotic probability distribution ideas which are central to this work, and it anticipates some of the theoretical results which follow.

Section 6 resumes the theoretical development. Its result is an expression for the components of the unique asymptotic probability distribution produced by the stationary algorithm variants which implement the mutation operator with nonzero mutation probability (i.e. the stationary two and three-operator algorithm variants). The result is expressed in terms of Cramer's Rule and thus its solution requires evaluation of determinants. The determinants are the characteristic polynomials, evaluated at $\lambda = 1$, of matrices derived from the state transition matrix produced in Section 4 by zeroing one row. A later section attacks the problem of explicitly solving the system, based upon the highly symmetrical nature of the state transition matrix, but some very significant results are obtainable from the product of Section 6 without explicit solution.

An essential step in establishing a connection between simulated annealing and the genetic algorithm is demonstrating the existence of a stationary distribution limit for the algorithm as the mutation probability approaches zero. Section 7 accomplishes that task and also provides a foundation for deducing, in Section 8, a mutation probability bound analogous to the annealing schedule bounds of the simulated annealing algorithm. The results developed in Sections 7 and 8 apply to both the two and three-operator algorithm variants.

A somewhat surprising result produced in Section 7 and anticipated by the empirical study reported in Section 5 is that the stationary distribution zero mutation probability limit does not necessarily isolate globally optimal solutions. In fact, it provides nonzero probability for all solutions of the underlying optimization problem and consequently the extrapolation of the simulated annealing methodology is less than exact. However, both

the empirical results presented in Section 5 and some results developed later in Section 9 suggest that the required limiting behavior can be approached as closely as desired by adjusting the algorithm parameters appropriately.

Section 9 attacks the problem of explicitly solving the system which results from the Cramer's Rule formulation of the stationary distribution of the time-homogeneous two and three-operator algorithms. It is a very extensive development which yields an expression for the coefficient of the general term in the Taylor's series expansion of the required determinants. It is based upon the highly symmetrical nature of the state transition matrix, as alluded to earlier.

The results of Section 9 are not reduced to a directly useable explicit solution. Nevertheless, they do provide significant insight into the functional form of the stationary distribution components. Furthermore, Section 9.5 points out some very significant identities which exist among the coefficients of the Taylor's series and suggests a method for continuing the Section 9 development based upon the algebra of symmetric and alternating polynomials. Explicit solution of the stationary distribution equations is the major incomplete task required for extrapolation of the simulated annealing convergence theory onto the genetic algorithm.

Section 10 summarizes this work and recapitulates the significant results. It also proposes continuation of two parts of this research: (1) pursuit of the stationary distribution solution and (2) refinement of the mutation probability control parameter bound.

An appropriate mathematical framework for examining both the simulated annealing and genetic algorithms is the theory of Markov chains. Appendix A is included to summarize some essential definitions and theorems. Appendix B is devoted to the Perron-Frobenius theorem, which is fundamental to the study of nonnegative matrices in general and Markov chains in particular. Several important Markov chain theorems are specializations of it and the key developments in Sections 6 and 7 require its application. All of the Appendix A and Appendix B results are provided without proof or elaboration,

but their foundation is obtainable from various references (e.g. [Cinl75] for the more elementary results in Appendix A, [Sene81] for the Appendix B material on the Perron-Frobenius Theorem and [Sene81, IsMa76] for the Appendix A ergodicity related definitions and theorems). These results are invoked freely in the following sections, either by specific reference to definition/theorem number, or if the context makes it appropriate, they are simply assumed.

Appendix C is provided as background for the Section 9.5 discussion on coefficient identities and extending the stationary distribution representation development. With the exception of Section C.4, the material presented in Appendix C is obtainable from advanced algebra texts (e.g. [MoSt64]). The symmetric/alternating polynomial generalization in Section C.4 is original.

Appendix D collects the computer program listings for the programs employed in generating the results reported in Section 5. The programs presented there were developed and executed on the Cray Y-MP operated by the Computer Science Directorate at Eglin AFB, Fl.

SECTION 2

SIMULATED ANNEALING

2.1 Overview

As noted in the introduction, a very commonly employed approach to the solution of nonconvex combinatorial optimization problems is a stochastic relaxation technique introduced by Kirkpatrick et al. and referred to as simulated annealing [KiGe83]. The technique is so named by virtue of its analogy to the annealing of solids, in which a crystalline solid is heated to its melting point and then allowed to cool very gradually until it is again in the solid phase at some nominal temperature. In the limiting case of infinitesimal cooling rate and absolute zero final temperature, the resulting solid achieves its most regular possible crystal lattice configuration (i.e. minimum lattice energy state), and hence is free of crystal defects. Simulated annealing establishes the connection between this sort of thermodynamic behavior and the search for the global minimum of an objective function in a combinatorial optimization problem, and further, it provides an algorithmic means of exploiting the connection. This section is a review of the technique with special emphasis on known results which bound the convergence behavior of computer algorithms belonging to the class.

2.2 Statistical Mechanics and Annealing of Solids

The fundamental assumption of statistical physics is that the thermodynamic behavior of a many particle system can be represented by a statistical ensemble, and that if the system is in thermal equilibrium, the time averages of macroscopic thermodynamic properties of the system are equal to the corresponding ensemble averages (ergodicity hypothesis). The random variable represented by the ensemble is the system thermal energy, and at thermal equilibrium the probability distribution is completely determined

by the system temperature. The distribution is known as the Boltzman distribution, or alternatively as the Gibbs distribution, and its form is

$$\Pr\{\overline{E} = E(i)\} = \frac{\exp\{-E(i)/kT\}}{Z(T)} \qquad \text{Eq. 2.1}$$

where $\overline{E}$ = the system thermal energy (a random variable)

$E(i)$ = the energy corresponding to state i

k = Boltzman's constant

T = the system temperature

$Z(T)$ = the partition function.

The factor

$$\exp\left\{\frac{-E(i)}{kT}\right\}$$

is called the Boltzman factor. The partition function provides the necessary normalization to make Eq. 2.1 a state occupancy probability. It can be expressed as

$$Z(T) = \sum_j \exp\left\{\frac{-E(j)}{kT}\right\}. \qquad \text{Eq. 2.2}$$

At elevated temperatures, the system represented by the probability distribution in Eq 2.1-2 occupies all states in its state space with nearly uniform probability, while at low temperatures, states having low energy are favored. When the temperature approaches absolute zero, only states corresponding to the minimum value of energy have nonzero probability. Thus, the thermodynamic system's energy function can be effectively searched for its minimum value by starting the system at an elevated temperature and allowing it to cool gradually to absolute zero, at which point one of its minimum energy states is occupied with probability one. This is the mechanism which guides the annealing of solids.

The cooling schedule employed in annealing solids is constrained by the requirement that the system be allowed to achieve thermal equilibrium at each temperature. The

Gibbs distribution only represents the system's energy distribution in the stationary case (i.e. equilibrium). If this requirement is not satisfied, defects can be frozen into the crystal lattice preventing the system from achieving the minimum possible energy state. This behavior is analogous to local minima entrapment in combinatorial optimization search. The restriction on the annealing schedule necessary to avoid it is the fundamental limitation on the annealing technique.

2.3 Combinatorial Optimization by Simulated Annealing

Simulated annealing approaches combinatorial optimization problems in a closely analogous fashion. In simulated annealing, the optimization problem's solution space corresponds to the state space of the analogous thermodynamic system and its cost function is analogous to the thermodynamic system's energy surface. The analog of the thermodynamic system's temperature is a nonnegative algorithm control parameter, T.

Two other algorithm components are also required. They are the stochastic next state generation and acceptance mechanisms, and they incorporate the dependence of the algorithm on the control parameter, T. The next state generation mechanism is employed by the algorithm to transform a current solution into a new candidate solution, and the acceptance mechanism is employed to decide whether to retain or discard the proposed new solution. Together, these stochastic operators are responsible for making the search algorithm simulate the thermodynamic system's statistical behavior. Consequently, they must satisfy certain requirements to assure coherent algorithm behavior. These requirements are explored in some depth later in the context of algorithm convergence behavior.

Conceptually, the operation of the simulated annealing algorithm can be described as follows. The algorithm starts at some initial value of the control parameter and with some initial solution. Then, the state generation mechanism is employed to synthesize a new candidate solution. The new solution is examined by the acceptance mechanism and either accepted or rejected. If it is accepted, the new solution becomes the current solution. Otherwise, the old current solution is retained. This process is repeated, generating a

sequence of temporary solutions, until an approximate equilibrium is achieved in which the solution space occupancy is described by the Gibbs distribution (Eq. 2.1-2). Once this approximate equilibrium is achieved, the control parameter value is reduced and the solution sequence is extended until equilibrium is achieved at the new control parameter value. This process is repeated until some termination condition (e.g. minimum control parameter value) is attained. The current solution at termination is then accepted as the solution to the optimization problem.

It is noted in passing that simulated annealing always involves minimizing a cost functional, never maximizing a reward. However, this causes no loss of generality because any combinatorial optimization problem can be translated into an equivalent minimization problem.

2.4 Theoretical Foundations of Simulated Annealing

The evolution of the search sequence of a simulated annealing algorithm as outlined, in which each succeeding solution in the sequence is determined stochastically based upon the current solution, suggests that the algorithm behavior can be described as a Markov chain. Indeed it can, and all of the known convergence results for simulated annealing algorithms are derived from analysis of Markov chain models [LaAa87, GeGe84, LuMe86, MiRo85, Rior58]. This subsection establishes a Markov chain model to represent the simulated annealing algorithm and then employs it in reviewing the development of the published convergence bounds. This development essentially follows [LaAa87].

2.4.1 A Markov Chain Model of Simulated Annealing

Let a combinatorial optimization problem be represented by the pair (S, C) where S is the problem's solution space and C is its cost function, and assume without loss of generality that the optimization problem requires minimization of C. Also, assume that S is finite. Then, a simulated annealing algorithm for solving this problem can be characterized by the quadruple $(S, i_0, \overline{P}_T, \tau)$ where S is as defined above and where $i_0 \in S$

is an initial candidate solution, $\overline{P}_T$ is a stochastic matrix which describes a stochastic state transition mechanism (the composition of the next state generation and acceptance mechanisms discussed in Section 2.3) and $\tau = \{T_k\}$ is a finite length monotone nonincreasing sequence of positive control parameter values. The first parameter value in τ is T_0 and the final value is T_f. $\overline{P}_T$ incorporates the algorithm dependence on both C and T.

The algorithm generates a sequence of candidate solutions, $\{i_k : 0 \le k \le f\}$, by employing the state transformation mechanism (described by $\overline{P}_T$) to transform solution i_k into i_{k+1}. At the $k^{\underline{th}}$ transition, $\overline{P}_T$ is completely determined by T_k. The solution sequence is extended until $T = T_f$, at which point the current solution, i_f, is accepted as the solution to the combinatorial optimization problem. Thus T_f signals algorithm termination. T_f can be allowed to depend on $\{i_k\}$ provided due regard is paid to the requirement for termination. Since the solution state transition mechanism is stochastic, and since the conditional dependence of the solution sequence only extends to one transition, the solution sequence is a Markov chain by Definition A1. Its state transition matrix is $\overline{P}_T$ (Definition A4).

The state transition matrix is decomposed into two parts for convenience in the following. It consists of the next state generation mechanism, $G_{ij}(T)$, which describes the probability of generating state j given that the current state is i, and the state acceptance mechanism, $A_{ij}(T)$, which describes the probability of accepting the generated state. Thus, $P_T(i,j)$ is written as

$$P_T(i,j) = \begin{cases} G_{ij}(T)A_{ij}(T) & j \neq i \\ 1 - \sum_{l=1, l \neq i}^{N} G_{il}(T)A_{il}(T) & j = i \end{cases} . \qquad \text{Eq. 2.3}$$

In this result, N = card(S) represents the cardinality of the solution space.

It is noted in passing that the usual form of the state acceptance mechanism is the so called Metropolis criterion [Metr53], given by

$$A_{ij}(T) = \min\left\{1, \exp\left[\frac{-(C(j)-C(i))}{T}\right]\right\}. \qquad \text{Eq. 2.4}$$

This is the form employed by Kirkpatrick et al. in the original work [KiGe83] and most others published since are variations of it. Also, the usual form of the next state generation mechanism is

$$G_{ij}(T) = G_{ij} = G_{ji} = \begin{cases} \frac{1}{N_i} & j \in S_i \\ 0 & \text{otherwise} \end{cases} \qquad \text{Eq. 2.5}$$

where $S_i \subset S$ is the set of states accessible from state i in one transition (by definition, $i \notin S_i$), and where $N_i = \text{card}(S_i)$. Note that G_{ij} defined by Eq. 2.5 is symmetric and independent of T.

2.4.2 Asymptotic Convergence Behavior

The subject of interest in the remainder of this section is a set of sufficient conditions on $\overline{P}_T$ and τ to ensure that an optimal solution is achieved. These conditions will prove to guarantee asymptotic convergence only (i.e. τ must be an infinite sequence, which of course violates the termination requirement of the algorithm). Two cases will be examined. The first only involves time-homogeneous (stationary) Markov chains (Definition A5) and is presented due to its relative ease of analysis. Its purpose is to provide a foundation for the essential ideas involved in the second case, which requires an appeal to ergodicity theorems for inhomogeneous (nonstationary) Markov chains. The useable convergence behavior results which are the goal of this effort derive from analysis of the second case.

The first (simple) algorithm is represented as a sequence of solutions evolving as a sequence of distinct Markov chains. Each Markov chain in the sequence executes at a fixed control parameter value (and hence is time-homogeneous) and each succeeding Markov chain executes at a lower (but strictly positive) parameter value. Thus, in the sequence τ, each distinct parameter value, T_l, is associated with a distinct time-homogeneous Markov chain and T_l occurs at some large number of consecutive locations, K_l, in τ. This case is hereafter referred to as the homogeneous (or stationary) algorithm.

The analysis of the convergence behavior of the homogeneous algorithm includes the hypothesis that each Markov chain in the sequence achieves its stationary distribution. This hypothesis is equivalent to $K_l \rightarrow \infty$ for all l (Definition A10, Theorem A3, Theorem A4).

In the second case, the algorithm is represented as a sequence of solutions evolving as a single inhomogeneous (nonstationary) Markov chain. This formulation is hereafter referred to as the inhomogeneous (or nonstationary) algorithm. In the inhomogeneous algorithm, the control parameter value is allowed to decrease (though not necessarily required to) after each state transition. The dependence of $G_{ij}(T)$ and $A_{ij}(T)$ on T results in the inhomogeneous behavior.

2.4.2.1 The Homogeneous Algorithm

In the homogeneous algorithm, the means of establishing the requirements for asymptotically optimal convergence is to first establish sufficient conditions for existence of the stationary distribution of each Markov chain and then to establish sufficient conditions to ensure that the stationary distribution converges to a uniform distribution over the set of optimal solutions as the control parameter value approaches zero. That is

$$\lim_{T \rightarrow 0} q_T(i) = \begin{cases} \dfrac{1}{N_{opt}} & i \in S_{opt} \\ 0 & \text{otherwise} \end{cases} \qquad \text{Eq. 2.6}$$

where $\bar{q}_T$ is the stationary distribution of the Markov chain executing at control parameter value T, $S_{opt} \subset S$ is the set of solutions $i \in S{:}C(i) = C_{opt}$ and $N_{opt} = \text{card}(S_{opt})$.

Theorems A1-A3 can be employed to deduce sufficient conditions on $P_T(i,j)$ (or alternatively on $G_{ij}(T)$ and $A_{ij}(T)$) to ensure the existence of the stationary distribution of each Markov chain in the sequence representing the homogeneous algorithm. Since only combinatorial (finite solution space) optimization problems are under consideration and since by definition the homogeneous algorithm only employs time-homogeneous Markov chains, the finite state space and time-homogeneity requirements of Theorem A3 are

satisfied. Beyond these requirements, existence of the stationary distribution of each Markov chain in the homogeneous algorithm only requires that the chain produced by $\overline{P}_T$ be irreducible and aperiodic (Definitions A7 and A9).

If $A_{ij}(T)$ is selected as the Metropolis criterion, Eq. 2.4, then

$$\forall i,j \in E, \forall T > 0 : A_{ij}(T) > 0.$$

Thus, from Eq. 2.3, the irreducibility requirement is transferred to the next state generation mechanism, $G_{ij}(T)$. Note that from Theorem A1, irreducibility can readily be achieved within the definition supplied by Eq. 2.5. Also, in [MiRo85], Theorem A2 is used to show that a sufficient condition for aperiodicity is

$$\forall T > 0 \; \exists i,j \in E \ni A_{ij}(T) < 1.$$

This condition is satisfied by the Metropolis criterion provided the trivial case indicated by

$$\forall i,j \in E : C(i) = C(j) = C_{opt}$$

is excluded, because then k,l always exist such that

$$C(l) = C_{opt} < C(k). \qquad \text{Eq. 2.7}$$

The sufficient condition on $A_{ij}(T)$ can then be met by selecting $i = l$ and $j = k$. (Use Eq. 2.7 in Eq. 2.4).

Although existence of the stationary distribution (or at least sufficient conditions on $G_{ij}(T)$ and $A_{ij}(T)$ to ensure its existence) are now established, and examples of G_{ij} and A_{ij} which meet these conditions provided, actually achieving the stationary distribution is only guaranteed after an infinite number of state transitions. This is equivalent to the thermal equilibrium constraint on the temperature schedule for annealing solids discussed in Section 2.2. Each Markov chain in the sequence representing the homogeneous algorithm is subject to this requirement, and consequently must be of infinite length.

Next, sufficient conditions to assure convergence of the stationary distribution of the final Markov chain in the homogeneous algorithm to the desired optimal distribution

(Eq. 2.6) are established. First, note that if the stationary distribution of a Markov chain in the sequence exists, then a function g(C(i), T) corresponding to that Markov chain exists such that

$$\forall i \in E : q_T(i) = \frac{g(C(i), T)}{\sum_j g(C(j), T)} \qquad \text{Eq. 2.8}$$

where g satisfies

$$\begin{aligned} &(1) \quad \forall i \in E, \forall T > 0 \quad : \quad g(C(i), T) > 0 \\ &(2) \quad \forall j \in E \quad : \quad \sum_{i \neq j} g(C(i), T) G_{ij}(T) A_{ij}(T) = g(C(i), T) \sum_{i \neq j} G_{ij}(T) A_{ij}(T) \end{aligned} \qquad \text{Eq. 2.9}$$

This can be deduced by noting that the uniquely determining conditions on $\bar{q}$ expressed in Theorem A3 are met by g satisfying Eq. 2.8 and 2.9. Eq. 2.9 is called the global balance equation. Close examination reveals that it is exactly the necessary condition for equilibrium state occupancy. A more restrictive condition, in which the balance holds for every pair of states on a pair-wise basis is called the detailed balance equation.

It can be shown that the following additional constraints on g guarantee convergence of the stationary distribution to the optimal (i.e. to Eq. 2.6) [MiRo85]. Note that Eq. 2.10(2) requires an exponential form.

$$\begin{aligned} &(1) \quad \lim_{T \to 0} g(A, T) = \begin{cases} 0 & A > 0 \\ \infty & A < 0 \end{cases} \\ &(2) \quad \frac{g(A_1, T)}{g(A_2, T)} = g(A_1 - A_2, T) \\ &(3) \quad \forall T > 0 : g(0, T) = 1 \end{aligned} \qquad \text{Eq. 2.10}$$

Collectively, Eq. 2.8-2.10 provide a set of sufficient conditions on $G_{ij}(T)$ and $A_{ij}(T)$ to assure convergence of the stationary distribution to Eq. 2.6. The key condition, the global balance equation, is implicit however, and thus is very difficult to apply. Nevertheless, it can be shown [LaAa87] that if $G_{ij}(T)$ and $A_{ij}(T)$ defined by Eq. 2.4 and Eq. 2.5 are

employed, the conditions are satisfied, and that the corresponding stationary distribution is provided by

$$\forall i \in E : q_T(i) = \frac{\exp\{-(C(i) - C_{opt})/kT\}}{\sum_j \exp\{-(C(j) - C_{opt})/kT\}} \quad . \qquad \text{Eq. 2.11}$$

The key to that development is that the $G_{ij}(T)$ and $A_{ij}(T)$ of Eq. 2.4 and 2.5 satisfy the detailed balance equation, the symmetry of G_{ij} being a critical consideration.

The behavior required by Eq. 2.10(1) is limiting behavior as $T \rightarrow 0$. Thus, these conditions assure convergence to the global minimum with probability one (i.e. convergence of the stationary distribution to Eq. 2.6), only if the sequence of Markov chains is infinite and $\lim_{l \rightarrow \infty} T_l = 0$. Recalling that a guarantee of achieving the stationary distribution requires that each Markov chain be of infinite length, the homogeneous algorithm is seen to require a doubly infinite sequence of solutions composed of an infinite sequence of infinitely long Markov chains.

2.4.2.2 The Inhomogeneous Algorithm

The behavior of the homogeneous algorithm, which requires that an infinite number of transitions be executed at each control parameter value, clearly is not very useful. The following reviews two published convergence results which extend the ideas developed for the homogeneous algorithm to the inhomogeneous counterpart [GeGe84, MiRo85]. These results adopt the sufficient conditions on $G_{ij}(T)$ and $A_{ij}(T)$ developed for the homogeneous algorithm as a starting point (i.e. irreducibility, aperiodicity and Eq. 2.8-2.10) and extend them to the case in which each time-homogeneous Markov chain is finite length (i.e. to the inhomogeneous algorithm). The key products of this effort are lower bounds on the algorithm control parameter's approach to zero. In both cases discussed here, the bound is of the form K/log(k) where k is the index of the Markov chain representing the inhomogeneous algorithm and K is independent of k. The following is a brief sketch of the approach taken to arrive at these results. It is common to both.

Given that $G_{ij}(T)$ and $A_{ij}(T)$ are selected as in Eq. 2.4 and 2.5, each state transition matrix in the inhomogeneous Markov chain of the inhomogeneous algorithm satisfies all of the sufficient conditions for stationary distribution existence and asymptotic convergence to optimality developed for the homogeneous algorithm (i.e. irreducibility, aperiodicity and Eq. 2.8-2.10). Further, the explicit form of the resulting stationary distribution is given by Eq. 2.11. Thus, for each transition matrix, $\overline{P}_{T_k}$), there exists an eigenvector, $\overline{q}_{T_k}$, having eigenvalue 1 and satisfying the probability vector conditions. Further, $\overline{q}_{T_k}$ converges to the limiting distribution of Eq. 2.6 as $T_k \rightarrow 0$. Consequently, Theorem A7 can be used to establish strong ergodicity (and hence the desired convergence behavior for $T_k \rightarrow 0$) provided (1) that weak ergodicity can be established and (2) that the inequality appearing in Theorem A7 obtains.

Under the hypothesis that $G_{ij}(T)$ and $A_{ij}(T)$ are defined in accordance with Eq. 2.4 and 2.5, in which case the required eigenvector is explicitly provided by Eq. 2.11, and that condition (1) (weak ergodicity) is satisfied, both [GeGe84] and [MiRo85] prove condition (2) of the above. The development is straightforward but tedious. Of more interest here is the means of establishing condition (1), because it leads to the annealing schedule bound.

Both developments employ Theorem A6 to establish weak ergodicity. The general approach is to use the definitions of $G_{ij}(T)$ and $A_{ij}(T)$, along with bounds on the extrema of either the cost function [GeGe84] or the slope of the cost function [MiRo85] to define bounds on the one step transition probabilities. The transition probability bound is then employed to arrive at an upper bound on the τ_1 coefficient of ergodicity of Theorem A5, which is used in turn in Theorem A6 to deduce a sufficient condition to guarantee weak ergodicity. The condition is in the form of a lower bound on the annealing schedule.

The first such result to be published is in [GeGe84]. The resulting bound is

$$T_k \geq \frac{N \times (C_{max} - C_{min})}{\log(k)} \qquad \text{Eq. 2.12}$$

$$k \geq 2$$

where C_{max} and C_{min} are the maximum and minimum values respectively of C(i) for $i \in S$ and $N = \text{card}(S)$. Thus, C_{min} is the desired C_{opt}.

The annealing schedule bound established in [MiRo85] is more refined than that of Eq. 2.12. It is given by

$$T_k \geq \frac{rL}{\log(k)} \qquad \text{Eq. 2.13}$$

$$k \geq 2$$

where r is the radius of the graph defining the accessible state neighborhoods of the next state generation mechanism (i.e. the $\{S_i\}$ where $S_i \subset S$ is defined in Eq. 2.5), and L is a constant which bounds the local slope of the cost function. Specifically, r and L are given by

$$r = \min_{i \in S - S_{max}} \max_{j \in S} d(i,j) \qquad \text{Eq. 2.14}$$

where d(i, j) is the distance of j from i, measured by the minimum number of state transitions required to arrive at j starting at i, where $S_{max} \subset S$ is the set of local maxima of C and

$$L = \max_{i \in S} \max_{j \in S_i} |\, C(j) - C(i) \,|\,. \qquad \text{Eq. 2.15}$$

Note that in the special case $S_i = S$ for all $i \in S$, then Eq. 2.14 and Eq. 2.15 reduce to $r = 1$ and $L = C_{max} - C_{min}$ respectively, and substitution into Eq. 2.13 yields

$$T_k \geq \frac{(C_{max} - C_{min})}{\log(k)}. \qquad \text{Eq. 2.16}$$

The Eq. 2.16 result is smaller than that of Eq. 2.12 by the factor 1/N.

Both of these published convergence results, as well as several others which are minor variations of them, are of the general form K/log(k). This behavior is the key limitation of the algorithm class, and is believed to be a fundamental limitation imposed by the neighborhood system inherent in the conventional simulated annealing state generation mechanism [GeGe84] (i.e. the fact that at low control parameter values, the likelihood of making the large state transition necessary to escape a local extremum is radically diminished). The simulated annealing literature includes some amount of speculation concerning state generation mechanisms which permit occasional large transitions even at low control parameter values.

SECTION 3

THE GENETIC ALGORITHM

3.1 Overview

The genetic algorithm is an iterative improvement stochastic search method appropriate for application to combinatorial optimization problems and based on the evolution of biological systems. It implements the fundamental idea of survival fitness on a population of string structures which are coded representations of solution candidates selected from the solution space of the optimization problem. The population of candidate solutions (which collectively represent the current estimate of the optimum solution) is subjected to a set of stochastic genetic operators which transform a current population into a new (descendent) population. A variety of distinct genetic operators (based on biological analogs) are available and are reported in the literature [Davi87, Gold89a, Gref85, Gref87]. The most important of them are (1) proportional reproduction, (2) crossover and (3) mutation. A one, two or three operator genetic algorithm employing combinations of these operators with fixed population size is referred to herein as a simple genetic algorithm.

The genetic operators are all implemented stochastically, but they do not result in a simple random walk through the search space. They represent a highly structured search which exploits the historical record of performance reflected at each stage of the search by the current population. It is the novel use of this historical record which is central to the appeal of the genetic algorithm.

Genetic algorithms usually operate on populations of bit-strings (i.e. the optimization problem is usually coded such that its search space is defined over a binary string alphabet), and they always attempt to maximize some strictly nonnegative objective

function. The evolution of the fixed size population of candidate solutions toward domination by optimal solutions is the algorithm goal.

The three genetic operators of a simple genetic algorithm are discussed in the next subsection. An analysis of their behavior requires introduction to the concept of schemata, or similarity templates, and that task is undertaken in a subsequent subsection. This section concludes with an assessment of the theoretical foundation available for the analysis of genetic algorithms.

3.2 The Simple Genetic Algorithm Operators

As noted above, the simple genetic algorithm employs three biologically inspired operators to transform each population of candidate solutions into a new (descendent) population. The following subsections examine each of these operators and how they influence the search evolution.

3.2.1 Reproduction

The genetic algorithm reproduction operator is the algorithmic analog of asexual reproduction. It is the means by which the objective function influences the evolution of the genetic algorithm search. It is implemented by evaluating each member of the current generation against the objective function and using the results to measure relative reproductive fitness (i.e. to provide a selection probability measure). Then, members of the current population are selected in accordance with this fitness measure to be members of the succeeding generation. This process is repeated (with statistically independent selection trials) until the entire new generation is populated.

In the absence of the other genetic operators, the reproduction operator tends to force the population to converge to the higher performing members of the current population. It eventually produces a uniform population. At any stage of the search (generation), only solutions which are represented by members of the current population can appear in any succeeding generation. In particular, no solution absent from the initial population is ever attainable. The reproduction operator exerts a strictly converging influence on the

search evolution. The other operators of the simple genetic algorithm circumvent this limitation in a controlled manner.

3.2.2 Crossover

The crossover operator in a genetic algorithm is the algorithmic analog of sexual reproduction. It produces the succeeding generation not by simply replicating the fittest members of the current generation but by mating the fittest members of the current generation to produce progeny with some of the "genetic" character of each parent. It is implemented by randomly exchanging parts of the strings representing the parents to produce descendent strings.

The crossover operator is implemented (with some given probability, p_c) after the reproduction operator has been invoked to select two reproducing parents. A string location is randomly selected (usually with uniform selection probability) and the parent bitstring on each side of the randomly selected location are exchanged to produce two progeny, which are then inserted into the succeeding population. This operation is repeated until the new generation is completely populated.

The crossover operator permits strings not represented in the current population to be generated in the succeeding population. That is, certain points in the solution space which are not represented in the current generation can be present in the successor generation. But the crossover operator is applied preferentially to high performance members of the current population, so it constitutes a judicious, informed tendency toward population divergence. This is the novel feature contributed by the crossover operator.

Even with the addition of crossover, the genetic algorithm search will eventually converge to a uniform population. In general the crossover operator causes a greater portion of the search space to be explored prior to convergence to uniformity, but for a given initial population, there are still unreachable points in the solution space. Further, even if a high performance solution is accessible from the initial population, some portion of the "gene pool" necessary to reach it can be irrevocably lost during the search evolution.

3.2.3 Mutation

The mutation operator is applied to each member of the successor generation created by the reproduction and crossover operators. It simply consists of randomly perturbing each descendent string with some (usually very small) perturbation probability, p_m. The operator exerts a diverging influence on the search algorithm, and it provides a means by which the search can, with some nonzero probability, always arrive at any point in the solution space. That is, no part of the "gene pool" is ever permanently extinguished if the mutation operator is implemented. Clearly, it is analogous to mutation in biological reproduction. Note also that if $p_m > 0$, the mutation operator precludes the algorithm from ever producing a permanently uniform population (i.e. it precludes algorithm convergence).

3.3 Building Blocks, Schemata and the Fundamental Theorem

The underlying premise of the genetic algorithm operators is that good solutions to an optimization problem over a bit-string solution space are composed of locally good substrings, and that assembling combinations of such locally good substrings is an effective way to search the space for globally good solutions. In the genetic algorithm literature, this is referred to as the building block hypothesis. For a problem to be amenable to genetic algorithm solution, this hypothesis should apply. In the genetics parlance, this hypothesis is stated as a requirement that the problem exhibit "...some but not too much epistasis" [Davi87]. The next subsection introduces an idea which helps to place this hypothesis on a more analytical basis, but the results are still incomplete.

3.3.1 Schema Defined

Let the solution space under consideration be the set of binary strings of length L, (i.e. $S = \{0,1\}^L$). Then, a schema (plural schemata), designated H, is a subset of S having the property that every member of H matches at some specified set of defining bit locations. Thus, if $L = 5$, then the schema H might be the set of length 5 bit-strings which match the string $(1,0,1,0,0)$ at the bit locations indicated by $H = \{s:s = (1,*,*,0,*)\}$, in

which the asterisks indicate "don't care" bits. The bit locations at which the schema is specified are the defining locations of the schema. The order of the schema, designated by o(H), is the number of its defining locations and can range from 0 to L. In this example, o(H) = 2. The defining length of the schema, designated δ(H), is the number of bit positions subtended by its outermost defining bit locations minus 1. In this example, $\delta(H) = 5 - 2 = 3$.

For a bit-string space of length L, there are exactly 3^L distinct schemata. This can be readily determined by noting that the distinct schemata are selected from $\{0, 1, *\}^L$. A given string selected from the space represents exactly 2^L distinct schemata. This results from the fact that the string is defined at all L bit positions, and hence is selected from $\{0, 1\}^L$. The schemata of an optimization problem's search space are the building blocks from which good solutions are to be constructed.

3.3.2 Schema Processing and the Fundamental Theorem

Let the constant population size of a simple genetic algorithm be designated M. Then, each generation produced by the algorithm represents some number, N, of distinct schemata that is bounded as follows

$$2^L \leq N < M \times 2^L. \qquad \text{Eq. 3.1}$$

The lower bound obtains when all M members are identical, and the upper bound represents a limit on schema diversity supported by the specified population size.

Now, briefly recalling the mechanisms implemented by the three simple genetic operators, it is possible to begin understanding their influence on the search evolution. In particular, the reproduction operator tends to reduce, never increase, the number of distinct schemata present in succeeding generations by selectively reproducing strings which are realizations of above average fitness schemata to the exclusion of below average competitors at the same set of defining locations.

The crossover operator, on the other hand, tends to produce new schemata by assembling high performance low order schemata in new combinations at the expense of

disrupting high order high performance schemata. The extent of population divergence introduced by the crossover operator is determined in part by the degree of schema diversity present in the current population. In particular, when the population becomes uniform, the crossover operator is nullified, because assembling substrings extracted from identical strings produces identical progeny.

The mutation operator also provides a disruptive mechanism which resists the converging influence of the reproduction operator. Since any schema can be produced by mutation with nonzero probability, the permanent extinction of any of the 3^L possible distinct schemata is precluded.

These ideas are captured in the following inequality, which is referred to in the literature as the Fundamental Theorem of Genetic Algorithms. It relates the number of copies of a particular schema in the current generation to the expected number of copies of the same schema in the succeeding generation. This inequality is derived in [Gold89a] from relatively simple probability notions. The development is not repeated here.

$$E\{\, m(H,k+1) \,\} \geq m(H,k) \times \left[\frac{R(H)}{\overline{R}} \right] \times \qquad \text{Eq. 3.2}$$

$$\left[1 - p_c \times \frac{\delta(H)}{(L-1)} - p_m \times o(H) \right]$$

where $m(H,k)$ = number of occurrences of schema H in the population at generation k,

$E\{\}$ = expected value operator,

$R(H)$ = average objective function value (> 0) of all strings in the current population which are realizations of H,

$\overline{R}$ = the average objective function value of the current population.

Equation 3.2 is an inequality because it does not consider the accretion of the schema H contributed by crossover and mutation. It only accounts for the disruptive effects of these operators. A more thorough treatment can be found on pp 9-13 of [Gref87], but the result is too cumbersome to be of much analytical value.

Qualitatively, Eq. 3.2 suggests that low order schemata occurring in the current population contribute to succeeding generations in direct proportion to the product of their number in the current generation and their average performance relative to the other schemata competing for dominance of the same set of defining locations. Crossover and mutation tend to disrupt this converging influence, and the disruptive effect of crossover is directly proportional to the defining length of the schema in question.

In view of Eq. 3.2, the building block hypothesis might be restated as a characteristic of genetic algorithm amenable optimization problems. A GA amenable problem is one for which a near optimum solution can be achieved, with a relatively small expenditure of search effort, by assembling high performance, low order schemata into novel combinations. If the objective function is such that (nonlinear) contributions from combinations of bits spanning widely separate bit locations are appreciable (i.e. if the objective function depends heavily on large defining length schemata), then the problem is not likely to be suitable for solution by genetic algorithm. On the other hand, if the objective function depends predominantly on short defining length schemata, then sorting through promising combinations of realizations of those schemata is likely to isolate good (though not necessarily optimal) solutions. Accomplishing the required sorting efficiently is the task for which genetic algorithms are well suited.

3.4 An Assessment of the Genetic Algorithm Theoretical Foundation

The existing theoretical foundation for analysis of genetic algorithms includes the fundamental theorem of genetic algorithms (Eq. 3.2) originally enunciated by Holland [Holl75] and extended by Bridges and Goldberg [BrGo87], the Walsh function approach to computing schema fitness averages contributed by Bethke [Beth80] and

generalizations of it [Gold88, Gold89b, BrGo89], a result concerning selection of the optimal population size for the algorithm [Gold85] in terms of the solution space dimension and the examination of the properties which make a problem difficult for genetic algorithms (the so called minimal deceptive problem) [Gold87, Gold89b]. Also, both De Jong [Dejo75] and Goldberg/Segrest [GoSe87] employ Markov chain methodology accompanied by approximate numerical analysis to examine certain specific problems concerning finite length chain behavior (e.g. genetic drift in a binary allele genetic algorithm).

No complete theoretical model exists for describing the operation of the simple genetic algorithm executing on a specified optimization problem. The central theme of the work underlying this paper is an attempt to develop such a model based upon the asymptotic behavior of a Markov chain which represents the algorithm.

SECTION 4

A MARKOV CHAIN MODEL OF THE SIMPLE GENETIC ALGORITHM

4.1 Overview

From the discussion of the simple genetic algorithm operators in Section 3.2, it is clear that the sequence of populations generated by the algorithm when executing on a specified combinatorial optimization problem is a stochastic process (with finite state space), and further that the conditional dependence of each population in the sequence on its predecessors is completely described by its dependence upon the immediate predecessor population. Thus, the sequence is a Markov chain (Definition A1). In this section, a nonstationary Markov chain model of the simple genetic algorithm is developed for one, two and three-operator variants of the algorithm. The model is tailored to resemble that offered in Section 2.4.1 for simulated annealing. The one-operator genetic algorithm model implements proportional reproduction only, while the two-operator variant employs reproduction in combination with mutation. The three-operator algorithm implements reproduction, mutation and crossover. This model hierarchy is employed because it provides some degree of insight into the effect that each operator has on the nature of the state space of the resulting Markov chain.

Describing and analyzing the operation of the simple genetic algorithm is facilitated by assuming that the underlying optimization problem is defined over a bit-string solution space. This assumption is not essential and sacrifices very little generality. It is implemented throughout the following sections.

4.2 The Markov Chain Model

Let a combinatorial optimization problem be characterized by the pair (S,R) where $S=\{0,1\}^L$ and R is a strictly positive real valued reward function, and assume, with no

loss of generality, that the problem requires maximization of R. Also, let a simple genetic algorithm designed to execute on this problem have fixed population size M, let $i \in S$ be interpreted as an unsigned integer ($0 \le i \le 2^L - 1$), and let a generation be represented by $\overline{m} = (m(0), m(1), \cdots, m(2^L - 1))$ where m(i) = the number of occurrences of solution $i \in S$ in the population. Thus, in the parlance of combinatorial mathematics, $\overline{m}$ is a distribution of M nondistinct objects over $N = \text{card}(S) = 2^L$ bins [Hall67, Rior58], and the set of all such distributions, $S' = \{\overline{m}\}$, is a suitable representation of the simple genetic algorithm search space. The cardinality of S' is given by

$$N' = \text{card}(S') = \binom{M + 2^L - 1}{M} = \binom{M + N - 1}{M}. \qquad \text{Eq. 4.1}$$

Since both N and M are finite, so is N'.

Then, if $\overline{m}_0 \in S'$ is selected as an initial population, the simple genetic algorithm can be represented by the quadruple $(S', \overline{m}_0, \overline{P}_{\overline{Q}}, \Gamma)$ where $\overline{P}_{\overline{Q}}$ is a state transition matrix (analogous to $\overline{P}_T$ of the simulated annealing model) and $\Gamma = \{\overline{Q}_k\}$ is a finite length sequence of parameter vectors $\overline{Q}_k = (p_m(k), p_c(k))$. The algorithm parameters $p_m(k)$ and $p_c(k)$ are respectively the mutation and crossover probabilities. In the following sections, the mutation probability sequence is employed in a role analogous to absolute temperature in simulated annealing, and consideration is limited hereafter to monotone nonincreasing sequences. In general, the only limitation on the crossover probability sequence is that its values are probabilities. However, in all of the following, consideration is limited to constant crossover probability sequences.

The first parameter vector in Γ is $\overline{Q}_0$ and the final parameter vector is $\overline{Q}_f$. The solution evolves as a sequence $\{\overline{m}_k\}$ of states $\overline{m}_k \in S'$ in which the conditional dependence of $\overline{m}_{k+1}$ on the sequence history is equivalent to its conditional dependence on $\overline{m}_k$, and thus the solution sequence is a Markov chain. In general, the chain is inhomogeneous (Definition A5). In Section 4.3 it is shown to be time-homogeneous if the parameter vectors are constant. As with the simulated annealing algorithm model, exhausting the sequence of

control parameter values, Γ, signals algorithm termination, and $\overline{Q}_f$ can be allowed to depend on $\{\overline{m}_k\}$ provided the algorithm termination requirement is satisfied.

4.3 State Behavior of the Simple Genetic Algorithm

In each of the next three subsections, the state transition mechanism (and its effect on the nature of the solution sequence) which results from employing a specified combination of the genetic algorithm operators to the Markov chain model is examined. The first case consists of a one-operator algorithm which employs only the reproduction operator. The second is a two-operator algorithm which employs reproduction and mutation. Finally a three-operator algorithm which includes crossover with reproduction and mutation is examined.

Although it is most natural to describe the genetic operators in the order reproduction/crossover/mutation, the course adopted in Section 3.2, the following development proceeds most instructively if mutation is included with reproduction in the two-operator algorithm and crossover is deferred to the three-operator case. This is due to the fact that the mutation operator provides the essential state space modification required to make the Markov chains of the time-homogeneous two and three-operator algorithms irreducible (Definitions A7 and A8, Theorem A1), and consequently causes them to have unique stationary distributions (Theorem A3). The one-operator algorithm (proportional reproduction only) does not satisfy the irreducibility requirement for existence of a unique stationary distribution. (Neither does the algorithm variant which employs reproduction and crossover without mutation). A unique stationary distribution means that the asymptotic state occupancy probability of the time-homogeneous two and three-operator algorithms is completely determined by the algorithm parameters and objective function. It is independent of the starting state (initial population). Asymptotic independence of the starting state is a necessary (but not sufficient) condition on the zero mutation probability limit of the stationary distribution of the time-homogeneous algorithm for the inhomogeneous algorithm counterpart to avoid (asymptotically) local minima entrapment.

4.3.1 A One-Operator Algorithm (Reproduction)

In this subsection, the nature of the state transition matrix is examined for the case of no crossover or mutation (i.e. $\overline{Q}_k = (0,0)$ for $0 \le k \le f$). In this case, the conditional probability of selecting a solution $i \in S$ from a population described by the state vector $\overline{n} \in S'$ is (i.e. proportional reproduction)

$$\forall i \in S, \forall \overline{n} \in S' : P_1(i \mid \overline{n}) = \frac{n(i) \times R(i)}{\sum_{j \in S} n(j) \times R(j)} \qquad \text{Eq. 4.2}$$

where the subscript 1 indicates that the one-operator case is under consideration. Thus, the conditional probability of the successor generation described by $\overline{m}$ given that the present generation is described by $\overline{n}$ is a multinomial distribution, i.e.

$$\begin{aligned} \forall \overline{m}, \overline{n} \in S' : P_1(\overline{m} \mid \overline{n}) &= \frac{M!}{\prod_{i \in S} m(i)!} \times \prod_{i \in S} P_1(i \mid \overline{n})^{m(i)} \\ &= \binom{M}{\overline{m}} \times \prod_{i \in S} P_1(i \mid \overline{n})^{m(i)} \qquad \text{Eq. 4.3} \\ &= \binom{M}{\overline{m}} \times \prod_{i \in S} \left[\frac{n(i) \times R(i)}{\sum_{j \in S} n(j) \times R(j)} \right]^{m(i)} \end{aligned}$$

where again the subscript 1 distinguishes the one-operator case, where the symbol

$$\binom{M}{\overline{m}} = \frac{M!}{\prod_{i \in S} M(i)!} \qquad \text{Eq. 4.4}$$

designates the indicated multinomial coefficient and where by definition

$$n(i) = 0 \;\Rightarrow\; \left[\frac{n(i) \times R(i)}{\sum_{j \in S} n(j) \times R(j)} \right]^{m(i)} = \begin{cases} 1 & m(i) = 0 \\ 0 & m(i) > 0 \end{cases}. \qquad \text{Eq. 4.5}$$

The transition probability matrix of the Markov chain representing the one-operator algorithm is composed of the array of conditional probabilities defined by Eq. 4.3, i.e.

$$\overline{P} = [P_1(\overline{m} \mid \overline{n})]. \qquad \text{Eq. 4.6}$$

Since it is independent of the sequence index (i.e. the parameter vectors are constant), the one-operator Markov chain is time-homogeneous (Definition A5).

The set of states which represent uniform populations (i.e. the states $\bar{m}_A \in S_A' \subset S'$ in which one component is M and all others are zero) are absorbing states of the Markov chain, because for any such state, $P_1(\bar{m}_A \mid \bar{m}_A) = 1$ and Definition A6 applies. Since it follows from Eq. 4.2-3 that $\forall \bar{n} \in S' - S_A' : P_1(\bar{n} \mid \bar{n}) < 1$, there are exactly $N = 2^L$ absorbing states. The corresponding rows of $\bar{P}$ are given by

$$\forall \bar{n}_A \in S_A' : P_1(\bar{m} \mid \bar{n}_A) = \begin{cases} 1 & \bar{m} = \bar{n}_A \\ 0 & \bar{m} \in S' - \{\bar{n}_A\} \end{cases} . \qquad \text{Eq. 4.7}$$

Thus, for each state $\bar{n}_A \in S_A'$, the associated row of the state transition matrix (Eq. 4.6) contains 1 in the principal diagonal location and 0 elsewhere. It follows that the $N' \times 1$ probability vector $\bar{q}_{\bar{n}_A}$ (Definition A2) whose $\bar{n}_A \in S_A'$ component is 1 is a stationary distribution (Definition A10) of the one-operator Markov chain. It is not unique because any of the $N = 2^L$ such vectors satisfies the requirement, as does any vector of the form $\bar{q} = \sum_{\bar{n}_A \in S_A} \gamma_{\bar{n}_A} \bar{q}_{\bar{n}_A}$ where $\gamma_{\bar{n}_A} \geq 0$ and $\sum \gamma_{\bar{n}_A} = 1$.

The absorbing states preclude irreducibility (Theorem A1), so the Markov chain does not satisfy the requirements of Theorem A3. The chain is aperiodic (Definition A9) however, because $\forall \bar{m} \in S' : P_1(\bar{m} \mid \bar{m}) > 0$ so the period of all states is 1. Thus, all of the conditions of Theorem A3 except irreducibility are met by the one-operator Markov chain.

The expected number of transitions required to arrive in an absorbing state, $E\{k_A\}$, is finite. An upper bound on $E\{k_A\}$ is given by

$$E\{k_A\} \leq \left(\frac{M \times R_{max}}{R_{min}} \right)^{2M} < \infty \qquad \text{Eq. 4.8}$$

where R_{min} and R_{max} are the extreme values of R. (Recall that R is assumed strictly positive, so $R_{max} \geq R_{min} > 0$). Eq. 4.8 can be derived by defining $p_A(k)$ as the conditional probability of arriving in the set of absorbing states, S_A', on the $k^{\underline{th}}$ transition given that the $k^{\underline{th}}$ state is not absorbing, letting p_{min} be a lower bound on $p_A(k)$, and bounding the series for $E\{k_A\}$ as follows

$$\begin{aligned} E\{k_A\} &= \sum_k k \times p_A(k) \prod_{l=1}^{k-1} (1 - p_A(l)) \\ &\leq \sum_k k \times \prod_{l=1}^{k-1} (1 - p_A(l)) \\ &\leq \sum_k k \times (1 - p_{min})^{k-1} \\ &= 1/(p_{min})^2. \end{aligned} \qquad \text{Eq. 4.9}$$

Next, note that a suitable bounding value on p_{min} is

$$p_{min} = \left[\frac{R_{min}}{M \times R_{max}} \right]^M > 0. \qquad \text{Eq. 4.10}$$

The desired bound on $E\{k_A\}$ (Eq. 4.8) is then obtained by using Eq. 4.10 in Eq. 4.9.

It is noteworthy that the above absorbing state convergence result does not require any assumption on the range of $R_{max} - R_{min}$. Even when the objective function exerts zero selective pressure (i.e. $\forall i \in S: R(i) = R_{min} = R_{max}$), the finite population size still results in convergence to an absorbing state. In the genetics parlance, this tendency is referred to as genetic drift. It is responsible for the inevitable convergence of the one-operator simple genetic algorithm, as discussed in Section 3.2.1.

4.3.2 A Two-Operator Algorithm (Reproduction and Mutation)

In this subsection, the nature of the state transition matrix is examined when the mutation operator is applied with some probability in the range $0 < p_m(k) < 1$ (i.e. $\overline{Q}_k = (p_m(k), 0)$). Let $P_2(i \mid \overline{n})$ and $P_2(\overline{m} \mid \overline{n})$ be the conditional distributions of the two-operator algorithm corresponding to the one-operator distributions defined by Eq. 4.2-4.5. Then, $P_2(i \mid \overline{n})$ and $P_2(\overline{m} \mid \overline{n})$ must account for the effect of nonzero p_m. This can be

accomplished by expressing $P_2(i \mid \bar{n})$ as a sum over all j of the corresponding $P_1(j \mid \bar{n})$ times a factor which accounts for the probability of the collection of mutation events required to transform j into i. This probability can be expressed as $p_m^{H(i,j)}(1-p_m)^{L-H(i,j)}$ where $H(i,j) = H(j,i)$ is the Hamming distance of the pair i, j. That is, H is a function defined on $S \times S$ with values in $\{0, 1, 2, \cdots, L\}$. H(i,j) is the number of bits which must be altered by mutation to transform i into j and L - H(i,j) is the number of bits which must remain unaltered. Thus, $P_2(i \mid \bar{n})$ can be written as

$$\forall i \in S, \forall \bar{n} \in S' : P_2(i \mid \bar{n}) = \sum_{j \in S} p_m^{H(i,j)}(1-p_m)^{L-H(i,j)} \times P_1(j \mid \bar{n})$$

$$= (1-p_m)^L \sum_{j \in S} \left[\frac{p_m}{(1-p_m)} \right]^{H(i,j)} \times P_1(j \mid \bar{n}) \quad \text{Eq. 4.11}$$

$$= \frac{1}{(1+\alpha)^L} \sum_{j \in S} \alpha^{H(i,j)} \times P_1(j \mid \bar{n})$$

where

$$\alpha = \frac{p_m}{(1-p_m)} \quad \text{Eq. 4.12}$$

and

$$p_m = \frac{\alpha}{(1+\alpha)}. \quad \text{Eq. 4.13}$$

For p_m=0 or p_m=1, Eq. 4.11 includes the indeterminate form 0^0 in some terms. Thus, the admissible range of p_m is restricted to $0 < p_m < 1$, and consequently that of α is $0 < \alpha < \infty$. However, cases corresponding to $p_m > 1/2 \Leftrightarrow \alpha > 1$ are of no practical interest (they are less random than the case $p_m = 1/2 \Leftrightarrow \alpha = 1$), and some of the following developments restrict consideration to the range $0 < p_m \leq 1/2 \Leftrightarrow 0 < \alpha \leq 1$.

Substituting Eq. 4.2 into Eq. 4.11 yields

$$P_2(i \mid \bar{n}) = \frac{1}{(1+\alpha)^L} \sum_{j \in s} \alpha^{H(i,j)} \times \frac{n(j) \times R(j)}{\sum_{k \in S} n(k) \times R(k)}.$$

It is also straightforward to show that

$$\sum_{k \in S} n(k) \times R(k) = \frac{1}{(1+\alpha)^L} \sum_{j \in S} \sum_{k \in S} n(k) \times R(k) \times \alpha^{H(j,k)}.$$

Thus, $P_2(i \mid \bar{n})$ can be expressed as

$$P_2(i \mid \bar{n}) = \frac{\sum_{j \in S} n(j) \times R(j) \times \alpha^{H(i,j)}}{\sum_{j \in S} \sum_{k \in S} n(k) \times R(k) \times \alpha^{H(j,k)}} \qquad \text{Eq. 4.14}$$

$$= \frac{\sum_{j \in S} n(j) \times R(j) \times \alpha^{H(i,j)}}{(1+\alpha)^L \sum_{k \in S} n(k) \times R(k)},$$

and $P_2(\bar{m} \mid \bar{n})$ is multinomially distributed as follows

$$\forall \bar{m}, \bar{n} \in S' : P_2(\bar{m} \mid \bar{n}) = \frac{M!}{\prod_{i \in S} m(i)!} \times \prod_{i \in S} P_2(i \mid \bar{n})^{m(i)}$$

$$= \binom{M}{\bar{m}} \times \prod_{i \in S} P_2(i \mid \bar{n})^{m(i)} \qquad \text{Eq. 4.15}$$

$$= \binom{M}{\bar{m}} \times \frac{1}{(1+\alpha)^{ML}} \prod_{i \in S} \left[\frac{\sum_{j \in S} n(j) \times R(j) \alpha^{H(i,j)}}{\sum_{k \in S} n(k) \times R(k)} \right]^{m(i)}.$$

The transition probability matrix of the Markov chain representing the two-operator algorithm is composed of the array of conditional probabilities defined by Eq. 4.15, i.e.

$$\bar{P} = [P_2(\bar{m} \mid \bar{n})]. \qquad \text{Eq. 4.16}$$

Since the elements of $\bar{P}$ depend on α (and hence by Eq. 4.12 on $p_m(k)$), the two-operator Markov chain is generally not time-homogeneous. It is time-homogeneous if the mutation probability is fixed.

Eq. 4.14-4.16 for the two-operator simple genetic algorithm are analogous to Eq. 4.2-4.6 for the one-operator variant except that $P_2(i \mid \bar{n})$ is strictly greater than zero for all $\bar{n} \in S'$. Thus, the two-operator analog of Eq. 4.5 is not required. Also

$$\lim_{\alpha \to 0^+} P_2(i \mid \bar{n}) = P_1(i \mid \bar{n})$$

and Eq. 4.17

$$\lim_{\alpha \to 0^+} P_2(\bar{m} \mid \bar{n}) = P_1(\bar{m} \mid \bar{n}).$$

The rows of the state transition matrix corresponding to the one-operator absorbing states have an especially simple form. Let $i_A \in S$ be the solution represented in the absorbing state $\bar{n}_A \in S_A$'. Then, from Eq. 4.14,

$$P_2(i \mid \bar{n}_A) = \frac{M \times R(i_A) \times \alpha^{H(i,i_A)}}{(1+\alpha)^L \times M \times R(i_A)} \qquad \text{Eq. 4.18}$$

$$= \frac{\alpha^{H(i,i_A)}}{(1+\alpha)^L}.$$

Thus, from Eq. 4.15,

$$P_2(\bar{m} \mid \bar{n}_A) = \binom{M}{\bar{m}} \frac{\alpha^{\sum m(i) \times H(i,i_A)}}{(1+\alpha)^{ML}}. \qquad \text{Eq. 4.19}$$

Since the reward function, R, is strictly positive by hypothesis, and since $\forall i,j \in S : 0 \le H(i,j) \le L$, it follows that for α in the range $0 < \alpha \le 1$, then

$$\alpha^L \sum_{j \in S} n(j) \times R(j) \le \sum_{j \in S} n(j) \times R(j) \times \alpha^{H(i,j)} \le \sum_{j \in S} n(j) \times R(j),$$

and consequently from Eq. 4.14 that

$$\forall i \in S, \forall \bar{n} \in S' : \left(\frac{\alpha}{1+\alpha}\right)^L \le P_2(i \mid \bar{n}) \le \left(\frac{1}{1+\alpha}\right)^L. \qquad \text{Eq. 4.20}$$

Using Eq. 4.20 in Eq. 4.15 yields

$$\forall \bar{m}, \bar{n} \in S' : \binom{M}{\bar{m}} \left(\frac{\alpha}{1+\alpha}\right)^{ML} \le P_2(\bar{m} \mid \bar{n}) \le \binom{M}{\bar{m}} \left(\frac{1}{1+\alpha}\right)^{ML}. \qquad \text{Eq. 4.21}$$

From the lower bound in Eq. 4.21, the final requirement of Theorem A3 (irreducibility) is fulfilled and the Markov chain for the time-homogeneous two-operator simple genetic algorithm possesses a unique stationary distribution, $\bar{q}_\alpha$, given by

$$\forall \overline{m} \in S' : (1)\ q_\alpha(\overline{m}) > 0$$

$$(2)\ \overline{q}_\alpha^T \overline{1} = 1$$

$$(3)\ \overline{q}_\alpha^T \overline{P} = \overline{q}_\alpha^T \ .$$

Since the stationary distribution is by definition a left eigenvector of the state transition matrix (Definition A10), it follows from Eq. 4.15 and 4.16 that the asymptotic state probability distribution of the time-homogeneous two-operator algorithm is completely determined by the objective function and the algorithm parameters. It is independent of the starting state, $\overline{m}_0$.

4.3.3 A Three-Operator Algorithm (Reproduction, Mutation and Crossover)

The three-operator simple genetic algorithm corresponds to the case $\forall k{:}Q_k = (p_m(k), p_c(k))$ with both $p_m(k)$ and $p_c(k)$ nonzero. Results analogous to Eq. 4.14-4.21 for the two-operator case are obtainable by defining a new function which is similar in character to the Hamming distance function employed in Section 4.3.2 for the two-operator case. This subsection completes that generalization. The result only reflects the crossover operation implicitly, however it permits some very significant conclusions concerning bounding values of the three operator conditional probabilities.

The new function, $I(i,j,k,s)$, is defined over an ordered quadruple (i,j,k,s) where $i,j,k \in S$ and where $s \in \{0,1,\cdots,L\}$ is a bit-string location. The states $i,j \in S$ represent respectively the first and second parent strings selected at a particular crossover opportunity and $k \in S$ represents a possible descendent string. The bit-string location s is the location randomly selected by the crossover operator, and normally it is uniformly distributed over its range. Thus, I is defined on $S \times S \times S \times \{0,1,2,\cdots,L\}$ and it takes on values selected from $\{0,1\}$ depending upon whether the indicated crossover operation is or is not consistent. That is, I assumes the value one if the bit-string k is produced by crossing the bit-strings i and j at the site s, and zero otherwise.

In terms of this crossover operator function, the conditional probability of producing, via reproduction and crossover, a solution $k \in S$ given a current population described by $\bar{n} \in S'$ is

$$P_2'(k \mid \bar{n}) = p_c \times \sum_{i \in S} \sum_{j \in S} P_1(i \mid \bar{n}) \times P_1(j \mid \bar{n}) \times \frac{1}{L} \sum_{s} I(i,j,k,s)$$

$$+(1 - p_c) \times P_1(k \mid \bar{n}) \qquad \text{Eq. 4.22}$$

$$= p_c \times \frac{1}{L} \times \sum_{i \in S} \sum_{j \in S} \sum_{s=1}^{s=L} P_1(i \mid \bar{n}) \times P_1(j \mid \bar{n}) \times I(i,j,k,s)$$

$$+(1 - p_c) \times P_1(k \mid \bar{n})$$

where $P_1(i \mid \bar{n})$ is as defined as in Eq. 4.2 and where $P_2'(i \mid \bar{n})$ refers to the two-operator algorithm consisting of reproduction and crossover without mutation. This result assumes uniformly distributed crossover site selection.

The array of conditional probabilities $[P_2'(i \mid \bar{n})]$ plays a role in the three-operator simple genetic algorithm very analogous to the role played by the array $[P_1(i \mid \bar{n})]$ in the two-operator variant. In fact, the $[P_2'(i \mid \bar{n})]$ array can be used as counterparts of Eq. 4.2 to develop results exactly analogous to Eq. 4.3 and Eq. 4.6. Further, for $\bar{n} \in S_A'$, Eq. 4.22 reduces to

$$P_2'(k \mid \bar{n}_A) = P_1(k \mid \bar{n}_A), \qquad \text{Eq. 4.23}$$

and consequently this (fictitious) two-operator algorithm (reproduction and crossover) demonstrates the same sort of absorbing state behavior as the one-operator algorithm.

From Eq. 4.22, the three-operator conditional probabilities and state transition matrix are expressible as

$$P_3(i \mid \bar{n}) = \frac{1}{(1+\alpha)^L} \times \sum_{j \in S} \alpha^{H(i,j)} \times P_2'(j \mid \bar{n}), \qquad \text{Eq. 4.24}$$

$$P_3(\overline{m} \mid \overline{n}) = \frac{M!}{\prod_{i \in S} m(i)!} \times \prod_{i \in S} P_3(i \mid \overline{n})^{m(i)} \qquad \text{Eq. 4.25}$$

$$= \left(\frac{M}{\overline{m}}\right) \times \prod_{i \in S} P_3(i \mid \overline{n})^{m(i)}$$

and

$$\overline{P} = [P_3(\overline{m} \mid \overline{n})]. \qquad \text{Eq. 4.26}$$

These results are developed in a fashion analogous to Eq. 4.14-4.16. From them, it follows that the three-operator Markov chain is time-homogeneous if both the mutation and crossover probabilities are fixed. In general it is not time-homogeneous.

From Eq. 4.22, 4.24 and 4.25, it follows that

$$\lim_{\alpha \to 0^+} P_3(i \mid \overline{n}) = P_2'(i \mid \overline{n}) \qquad \text{Eq. 4.27}$$

and $$\lim_{\alpha \to 0^+} P_3(\overline{m} \mid \overline{n}) = P_2'(\overline{m} \mid \overline{n}).$$

Also, from Eq. 4.23-4.25, the three-operator analogs of Eq. 4.18-19 apply

$$P_3(i \mid \overline{n}_A) = \frac{\alpha^{H(i, i_A)}}{(1+\alpha)^L} \qquad \text{Eq. 4.28}$$

$$P_3(\overline{m} \mid \overline{n}_A) = \left(\frac{M}{\overline{m}}\right) \frac{\alpha^{\sum m(i) \times H(i, i_A)}}{(1+\alpha)^{ML}}. \qquad \text{Eq. 4.29}$$

Additionally, since

$$\alpha^L \sum_{j \in S} P_2'(j \mid \overline{n}) \le \sum_{j \in S} P_2'(j \mid \overline{n}) \times \alpha^{H(i,j)} \le \sum_{j \in S} P_2'(j \mid \overline{n}),$$

the three-operator analogs of Eq. 4.20-21 follow from Eq. 4.24-25, i.e.

$$\forall i \in S, \forall \overline{n} \in S' : \left(\frac{\alpha}{1+\alpha}\right)^L \le P_3(i \mid \overline{n}) \le \left(\frac{1}{1+\alpha}\right)^L \qquad \text{Eq. 4.30}$$

and

$$\forall \overline{m}, \overline{n} \in S' : \left(\frac{M}{\overline{m}}\right)\left(\frac{\alpha}{1+\alpha}\right)^{ML} \le P_3(\overline{m} \mid \overline{n}) \le \left(\frac{M}{\overline{m}}\right)\left(\frac{1}{1+\alpha}\right)^{ML}. \qquad \text{Eq. 4.31}$$

All of the state space characteristics described in 4.3.2 for the two-operator algorithm follow. In particular, the Markov chain of the three-operator algorithm is irreducible. Thus, a unique stationary distribution exists for the time-homogeneous three-operator simple genetic algorithm, and as in the two-operator case it is completely determined by the objective function and the algorithm parameter values.

4.3.4 Summary

The asymptotic behavior of the one-operator simple genetic algorithm is dominated by the states which correspond to uniform populations, the one-operator absorbing states. The algorithm necessarily arrives at some member of the absorbing state set within a finite number of algorithm iterations (Eq. 4.8). The asymptotic probability distribution depends upon the algorithm initial population, $\overline{m}_0$. This observation is equivalent to the fact, established in Section 4.3.1, that the stationary distribution of the one-operator algorithm is not unique.

A unique stationary distribution exists for the time-homogeneous two and three-operator algorithm variants (with $\alpha > 0$), or equivalently, their asymptotic probability distributions are independent of $\overline{m}_0$. However, in the $\alpha \rightarrow 0^+$ limit, both the two and three-operator algorithms degenerate into the absorbing state behavior which typifies the one-operator case (Eq. 4.17 and Eq. 4.23, 4.27). A very important question is whether the unique stationary distributions of the two and three-operator algorithms approach limits as $\alpha \rightarrow 0^+$. Section 7 answers that question affirmatively, and in Section 8, the lower bounds reflected in Eq. 4.21 and Eq. 4.31 are employed to arrive at a monotone decreasing sequence bound on $p_m(k)$ sufficient to guarantee that the limiting distribution is achieved (asymptotically) by the inhomogeneous two and three-operator Markov chains.

The analogous conditional probability arrays $[P_1(i \mid \overline{n})]$ and $[P_2'(i \mid \overline{n})]$, whose elements are defined by Eq. 4.2 and Eq. 4.22 respectively, play a very essential role in the following sections, especially in Section 9. Most of the results developed hereafter apply equally to the two and three-operator algorithm variants by substituting from these

conditional probability arrays appropriately. Thus, in much of the following, the notation modifiers are suppressed, so that the elements of either of these arrays are denoted by $P(i \mid \bar{n})$, with the specific array reference being determined by context.

SECTION 5

SOME EMPIRICAL RESULTS

5.1 Overview

This section reports the results of some computer simulations based upon the genetic algorithm Markov chain model developed in Section 4. Their purpose is to help fix some of the state space and asymptotic probability distribution ideas which are central features of this work.

The results reported here are separated into four subsections. Section 5.2 concerns enumeration of the state space, S'. Section 5.3 is devoted to generation of reward function data, which are subsequently used in the two remaining subsections. Section 5.4 illustrates the behavior of some selected conditional probabilities as a function of the algorithm control parameter, α. The results of the primary simulation task are reported in Section 5.5. They concern computation of the three-operator stationary distribution at extremely low (approaching zero) values of the mutation probability control parameter.

One of the significant theoretical results developed in subsequent sections is suggested by the data presented in Section 5.5. It is that the zero mutation probability limiting stationary distribution provides nonzero probability for all states corresponding to uniform populations (i.e. one-operator absorbing states), including those which represent suboptimal solutions. This result poses a complication for the attempt to extrapolate the simulated annealing convergence theory onto the genetic algorithm, as discussed further in section 5.5.

All simulation results included here were generated on the Cray Y-MP computer at the Eglin AFB, Fl. Computer Science Directorate. The data presented in Section 5.5 concerning the primary simulation task (the converged limiting stationary distribution

results) includes some CPU utilization statistics which reflect the approximately 180 hours of CPU time expended in generating that data. The source program listings for the programs employed in generating the results of this section are included in Appendix D.

5.2 State Space Enumeration

The results appearing in this section are of two primary types. The first is a table of computed state space cardinality values, N', at a variety of combinations of bit-string length, L, and population size, M. These results are products of the program GET_NPS.F appearing in Appendix D. It implements Eq. 4.1. The results are collected in Table 5-1.

In addition to the N' column, Table 5-1 includes a similar column labeled N''. It denotes the cardinality of a space designated S'' which is related to S' and whose significance is established in Section 9. Its cardinality is given by

$$N'' = \binom{M+N}{M}. \qquad \text{Eq. 5.1}$$

The data recorded in column N'' of Table 5-1 are computed from this equation by the program GET_NPS.F.

Table 5-1

State Space Cardinality

M	L	N	N'	N''
1	1	2	2	3
	2	4	4	5
	3	8	8	9
	4	16	16	17
	5	32	32	33
	6	64	64	65
	7	128	128	129
	8	256	256	257
2	1	2	3	6
	2	4	10	15
	3	8	36	45
	4	16	136	153
	5	32	528	561
	6	64	2080	2145
	7	128	8256	8385
	8	256	32896	33153

Table 5-1 (continued)

M	L	N	N'	N''
3	1	2	4	10
	2	4	20	35
	3	8	120	165
	4	16	816	969
	5	32	5984	6545
	6	64	45760	47905
	7	128	357760	366145
	8	256	2829056	2862209
4	1	2	5	15
	2	4	35	70
	3	8	330	495
	4	16	3876	4845
	5	32	52360	58905
	6	64	766480	814385
	7	128	11716640	12082785
	8	256	183181376	186043585
5	1	2	6	21
	2	4	56	126
	3	8	792	1287
	4	16	15504	20349
	5	32	376992	435897
	6	64	10424128	11238513
	7	128	309319296	321402081
	8	256	9525431552	9711475137
6	1	2	7	28
	2	4	84	210
	3	8	1716	3003
	4	16	54264	74613
	5	32	2324784	2760681
	6	64	119877472	131115985
	7	128	6856577728	7177979809
	8	256	414356272512	424067747649
7	1	2	8	36
	2	4	120	330
	3	8	3432	6435
	4	16	170544	245157
	5	32	12620256	15380937
	6	64	1198774720	1329890705
	7	128	131254487936	138432467745
	8	256	15508763342592	15932831090241
8	1	2	9	45
	2	4	165	495
	3	8	6435	12870
	4	16	490314	735471
	5	32	61523748	76904685

Table 5-1 (continued)

M	L	N	N'	N''
	6	64	10639125640	11969016345
	7	128	2214919483920	2353351951665
	8	256	509850594887712	525783425977953

The second set of state space enumeration results reported here is a listing of the elements of S' for a variety of values of M and L. The principal purpose of this particular simulation task is to verify the computer algorithm (and the implementing subprograms) employed to generate the state space vectors for use in the primary simulation task reported in Section 5.5. The data in Tables 5-2, 5-3 and 5-4 below are products of the program GET_SPS.F appearing in Appendix D. The tabulated results are representative of data produced for M and L ranging up to 5 and 4 respectively, for which N' = 15504. In each case generated, the cardinality of the result agreed with that predicted by Eq. 4.1 and recorded in Table 5-1.

Table 5-2

S' at M=2, L=2

2	0	0	0
1	1	0	0
1	0	1	0
1	0	0	1
0	2	0	0
0	1	1	0
0	1	0	1
0	0	2	0
0	0	1	1
0	0	0	2

Table 5-3

S' at M=3, L=2

3	0	0	0
2	1	0	0
2	0	1	0
2	0	0	1
1	2	0	0
1	1	1	0
1	1	0	1
1	0	2	0
1	0	1	1
1	0	0	2
0	3	0	0
0	2	1	0
0	2	0	1
0	1	2	0
0	1	1	1
0	1	0	2
0	0	3	0
0	0	2	1
0	0	1	2
0	0	0	3

Table 5-4

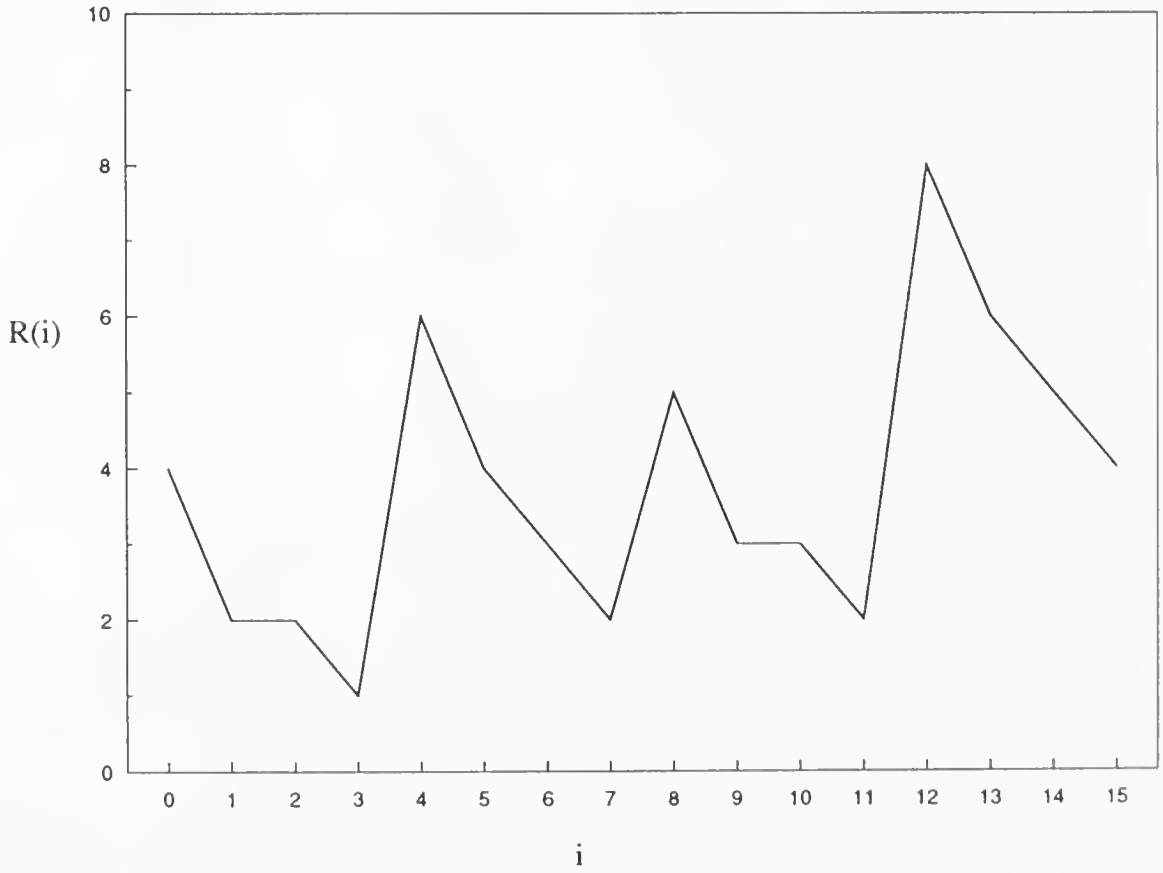
S' at M=2, L=3

2	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	0	1	0	0	0	0	0
1	0	0	1	0	0	0	0
1	0	0	0	1	0	0	0
1	0	0	0	0	1	0	0
1	0	0	0	0	0	1	0
1	0	0	0	0	0	0	1
0	2	0	0	0	0	0	0
0	1	1	0	0	0	0	0
0	1	0	1	0	0	0	0
0	1	0	0	1	0	0	0
0	1	0	0	0	1	0	0
0	1	0	0	0	0	1	0
0	1	0	0	0	0	0	1
0	0	2	0	0	0	0	0
0	0	1	1	0	0	0	0
0	0	1	0	1	0	0	0
0	0	1	0	0	1	0	0
0	0	1	0	0	0	1	0
0	0	1	0	0	0	0	1
0	0	0	2	0	0	0	0
0	0	0	1	1	0	0	0
0	0	0	1	0	1	0	0
0	0	0	1	0	0	1	0
0	0	0	1	0	0	0	1
0	0	0	0	2	0	0	0
0	0	0	0	1	1	0	0
0	0	0	0	1	0	1	0
0	0	0	0	1	0	0	1
0	0	0	0	0	2	0	0
0	0	0	0	0	1	1	0
0	0	0	0	0	1	0	1
0	0	0	0	0	0	2	0
0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	2

5.3 Reward Function Data

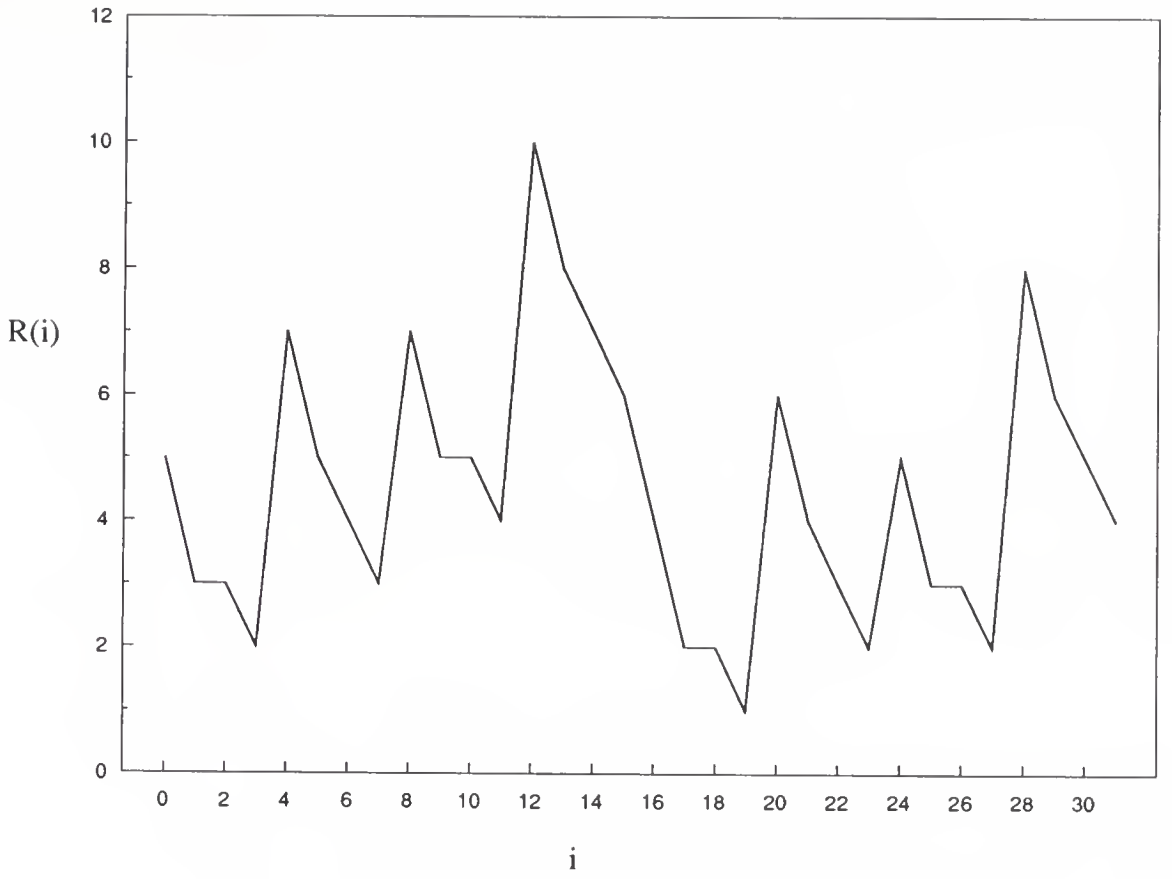
This section presents two sets of reward function data, one for a four-bit optimization problem and another for a five-bit version. Both data sets are products of the program GET_R.F provided in Appendix D. These data sets are employed in the simulations presented in Sections 5.4 and 5.5. Figure 5-1 presents the four-bit function and Figure 5-2 the five-bit version.

In both data sets, the solution state which maximizes the reward value is the $i \in S$ represented by the decimal integer value 12. That is, for the four-bit function, $i_{opt} = 1100$, and its five-bit counterpart is $i_{opt} = 01100$. The reward function value for the arbitrary $i \in S$ is then computed by assigning the value 1 for each length 0, 1 or 2 schema (Section 3.3.1) in agreement with the optimum bit pattern and summing the contributions. Thus, for example, for the four-bit reward function, the bit-string 0000 has function value 4, generated by summing the contributions from the single matching length 0 schema, two matching length 1 schemata and the one matching length 2 schema. A strictly positive reward function is guaranteed since every string matches the single length 0 schema.



Four-Bit Reward Function

Figure 5-1



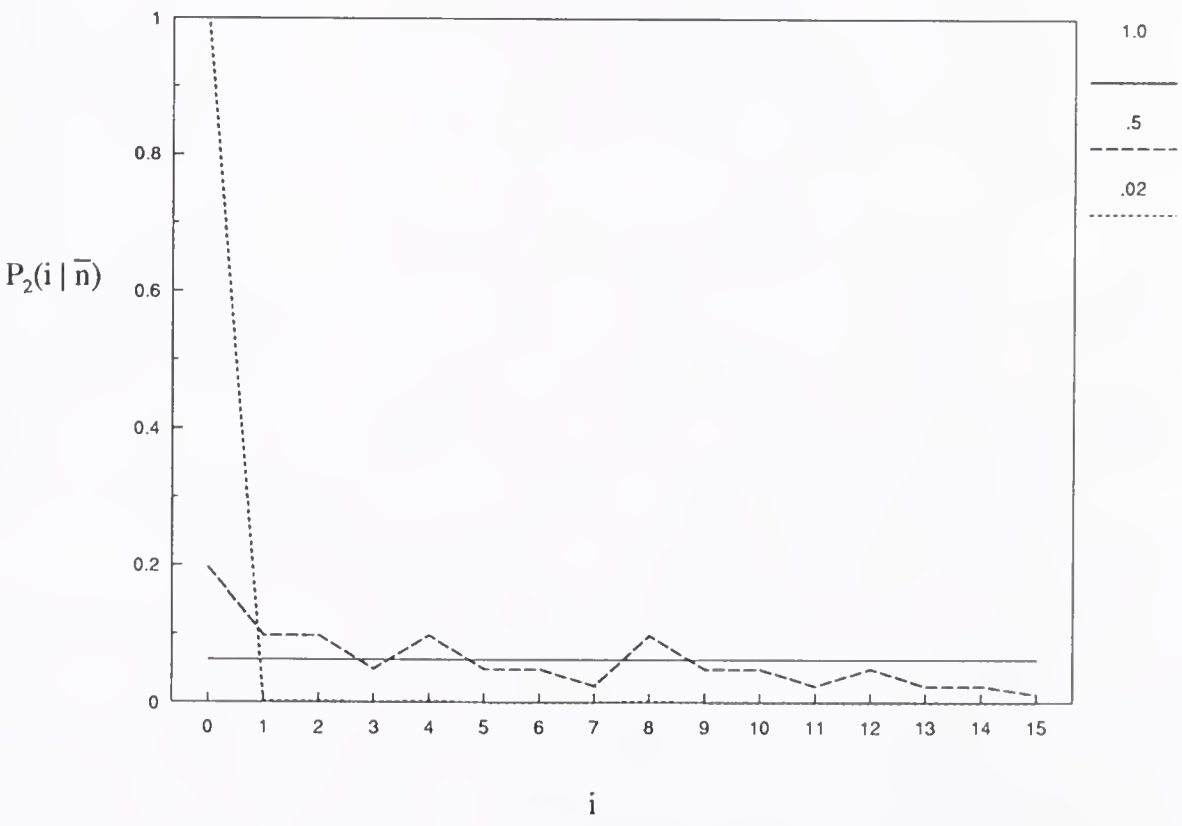
Five-Bit Reward Function

Figure 5-2

5.4 Conditional Probabilities Versus α

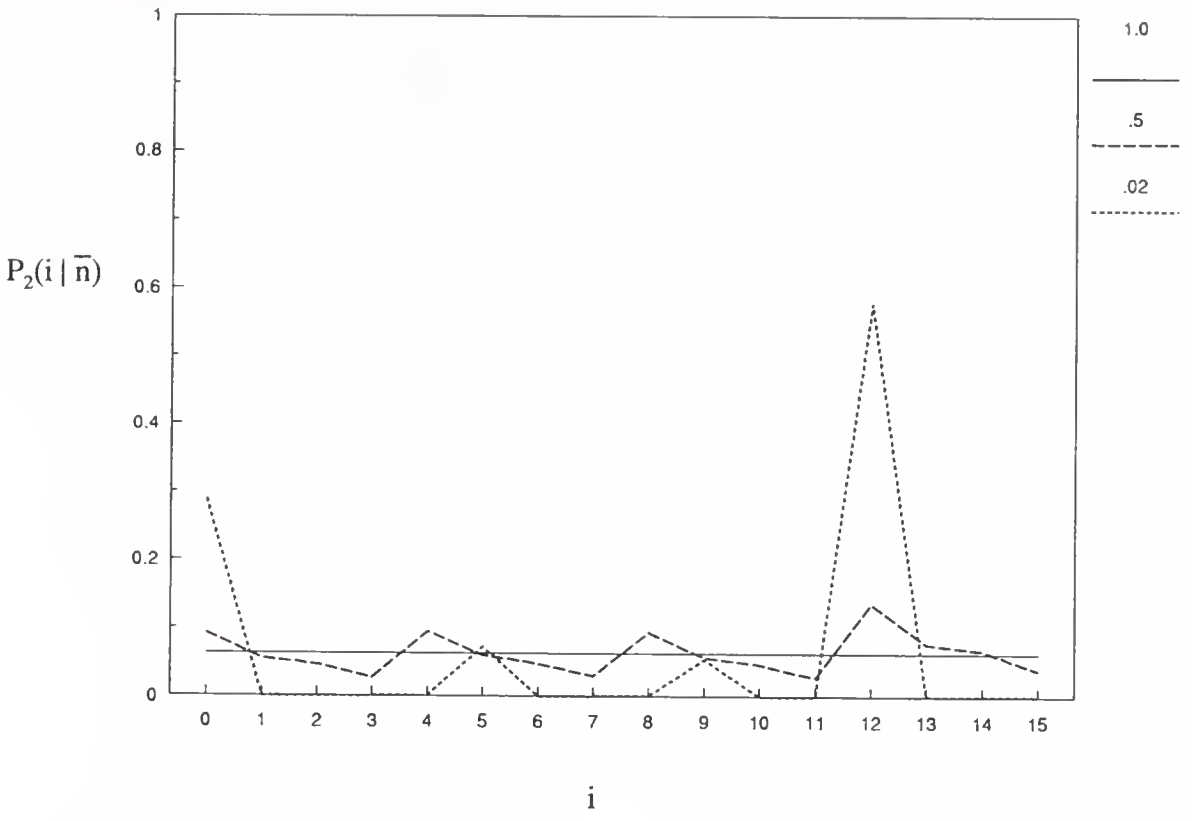
The following four figures present plots of two and three-operator conditional probabilities at two selected current states, $\bar{n}$. These results are computed from Eq. 4.14 and Eq. 4.24. The plots are generated for the four-bit problem with reward function given in Figure 5-1 and with M = 6. From Table 5-1, the cardinality of S' for these examples is N' = 54264. The conditional probabilities are provided at two selected $\bar{n}$ vectors, one representing the uniform population $\bar{n}$ = (6000000000000000) and one the mixed population state $\bar{n}$ = (2000010001002000), and at three values of the mutation probability parameter. The two and three-operator results are respectively products of the computer programs GET_P2INS and GET_P3INS provided in Appendix D.

The purpose of the tests from which these data are produced is verification of the computer algorithms (and the implementing subprograms) employed to generate the conditional probability calculations required by the primary simulation task reported in Section 5.5. Thus, for example, all conditional probability distributions are uniform at $\alpha = 1$ as is required by Eq. 4.14 and Eq. 4.24, and for $\alpha \rightarrow 0^+$ all conditional probability distributions approach the one-operator counterparts as is required by Eq. 4.17 and Eq. 4.27. Also, the two and three-operator conditional probabilities are identical for the uniform population case (Figures 5-3 and 5-5) as required by Eq. 4.18 and 4.28, and the three-operator mixed population state case allows generation of solutions not present in the current population even in the zero mutation probability limit.



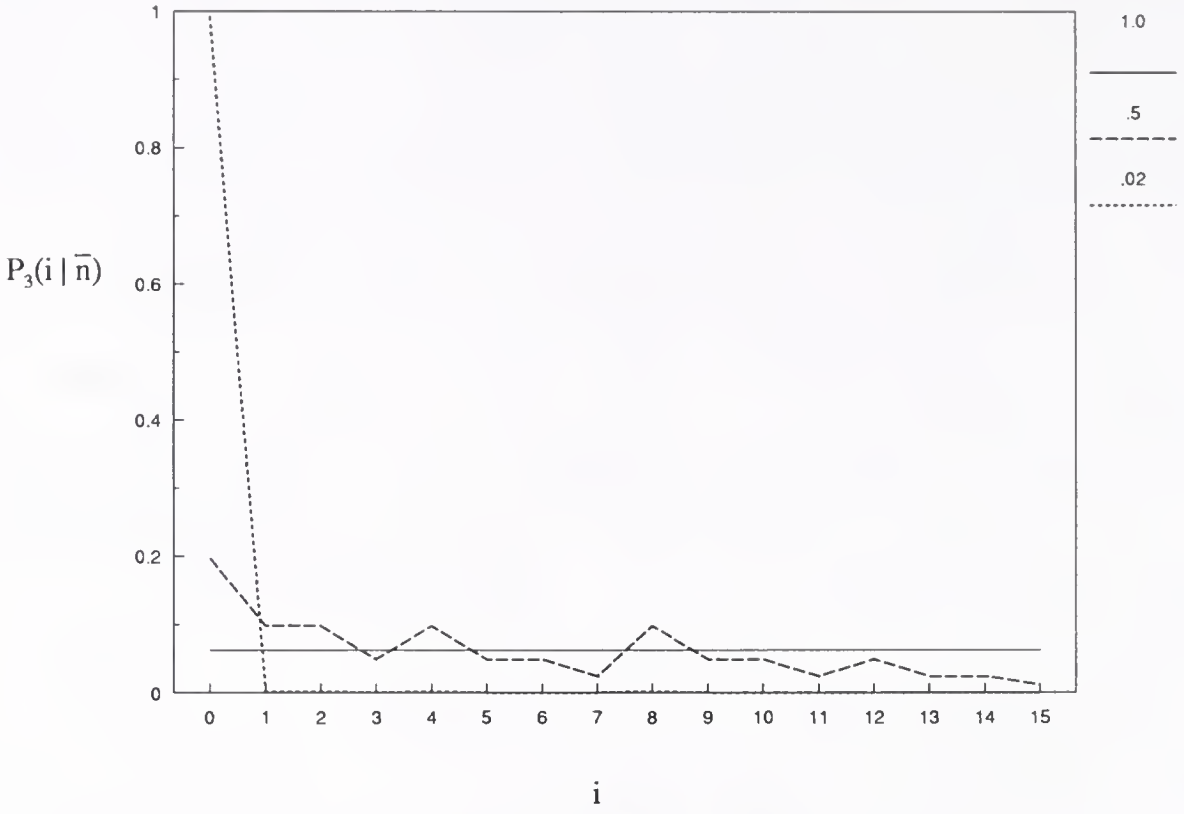
$P_2(i \mid \bar{n})$ at $\bar{n} = (6000000000000000)$

Figure 5-3



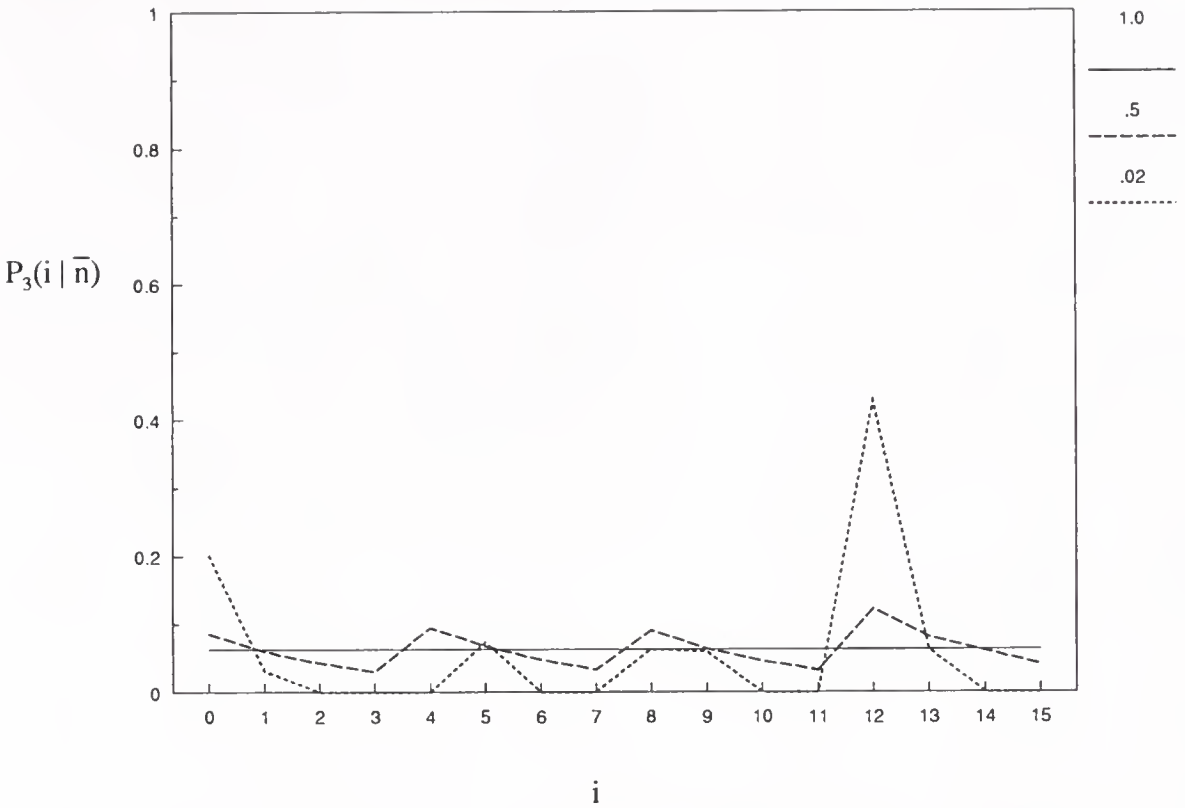
$P_2(i \mid \bar{n})$ at $\bar{n} = (2000010001002000)$

Figure 5-4



$P_3(i \mid \bar{n})$ at $\bar{n} = (60000000000000000)$

Figure 5-5



$P_3(i \mid \bar{n})$ at $\bar{n}$ = (2000010001002000)

Figure 5-6

5.5 Converged Limiting Stationary Distributions

The following data represent converged three-operator stationary distribution results for both four and five-bit problems at a variety of population sizes. The results recorded in Figures 5-7 through 5-16 are products of the computer program GET_3STAT.F included in Appendix D. They are obtained by repeatedly multiplying a current state probability vector by the three-operator state transition matrix until a termination criterion representing approximate convergence is attained. The starting probability vector is the multinomial distribution corresponding to a uniformly distributed $P_3(i \mid \bar{n})$ array, and the termination criterion is that the sum of the probabilities for all nonuniform population states is less than 0.004.

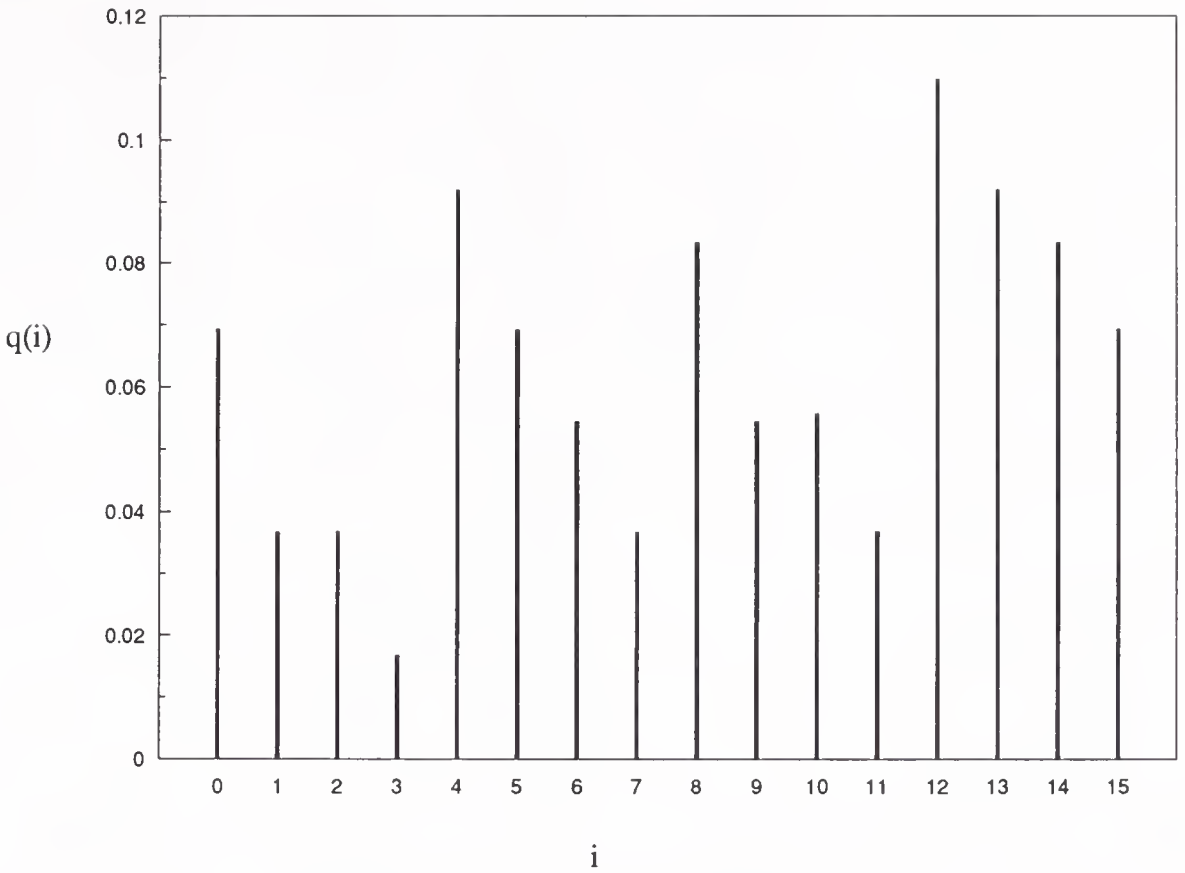
All of the results reported here are for extremely small α (approaching zero) and thus, as predicted by the model, only the states corresponding to uniform populations

(one-operator absorbing states) have nonzero probability. Consequently, only the final probabilities for the uniform population states are displayed in Figures 5-7 through 5-16, with each such state indexed by the decimal integer value corresponding to the solution represented.

Table 5-5 summarizes the Cray Y-MP computer resources expended in generating these data. Tabulated there are the number of vector multiplications (of dimension N') required to attain the termination condition and the CPU time utilized. The CPU time is in seconds, rounded to the nearest integer. The tabulated data are collected from the log files generated in the computer runs which produced the stationary distribution data for Figures 5-7 through 5-16.

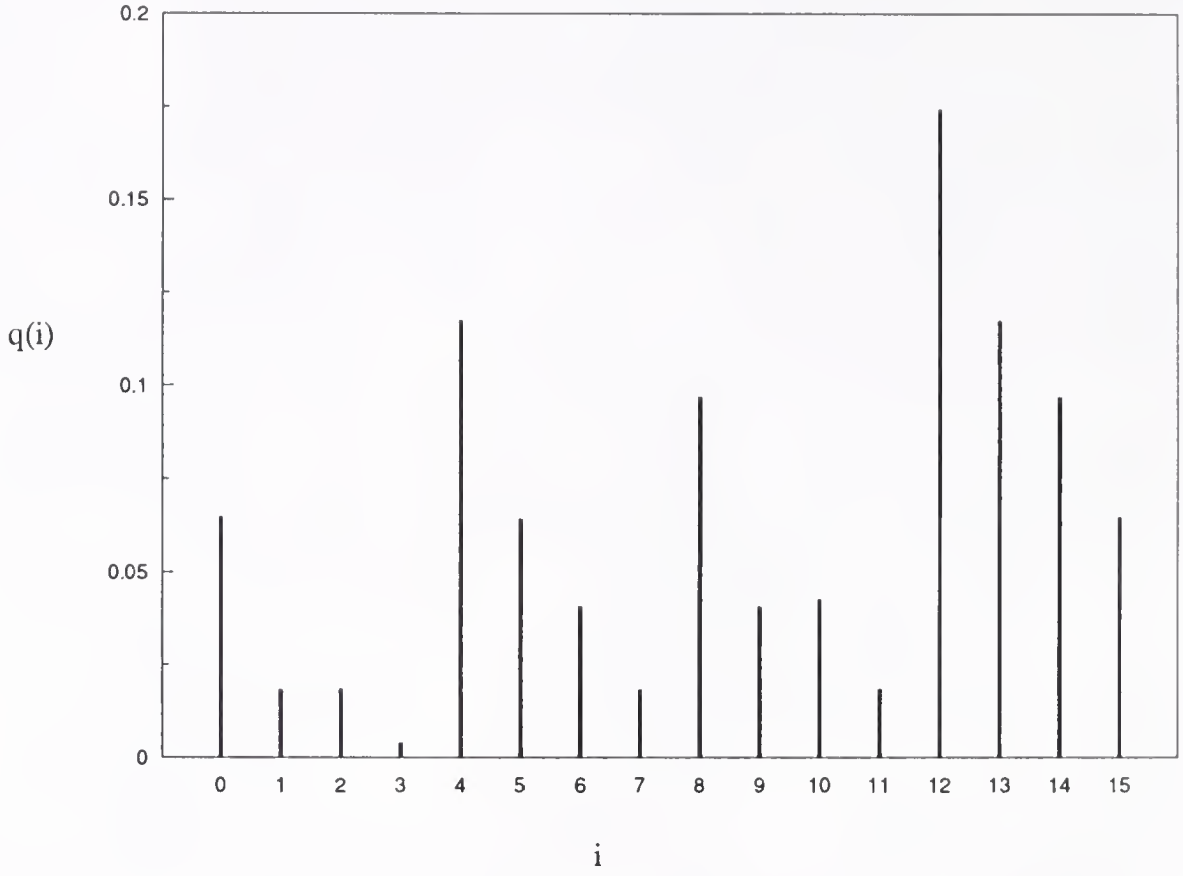
The limiting distribution entropy results in Figures 5-17 and 5-18 are computed from the converged stationary distributions. The results are recorded in bits and are plotted as a function of population size.

A very significant result suggested by the limiting stationary distribution data is that the $\alpha \rightarrow 0^+$ value of the stationary distribution is nonzero for all possible uniform states. This behavior, which is confirmed by theoretical results developed in Section 7, precludes extrapolation of the simulated annealing global optimality convergence result onto the genetic algorithm. However, as suggested by the data plotted in Figures 5-17 and 5-18, it may be possible to approach the desired limiting behavior as closely as required by adjusting the population size parameter. Those figures indicate that for sufficiently large values of the population size parameter, the limiting distribution is dominated by optimal solutions, and that the limiting distribution entropy decreases monotonically with increasing population size. Results developed in Section 9 reinforce this premise.



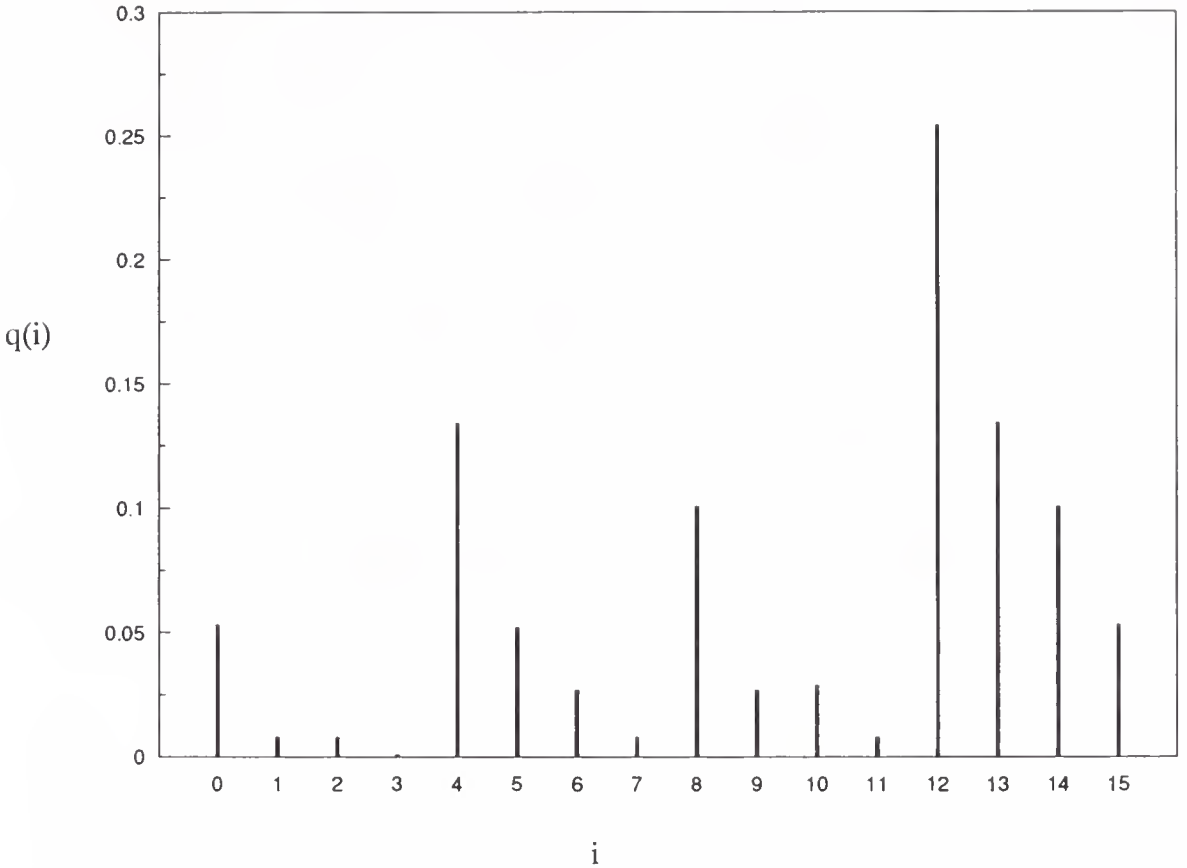
Limiting Stationary Distribution at M=2, L=4

Figure 5-7



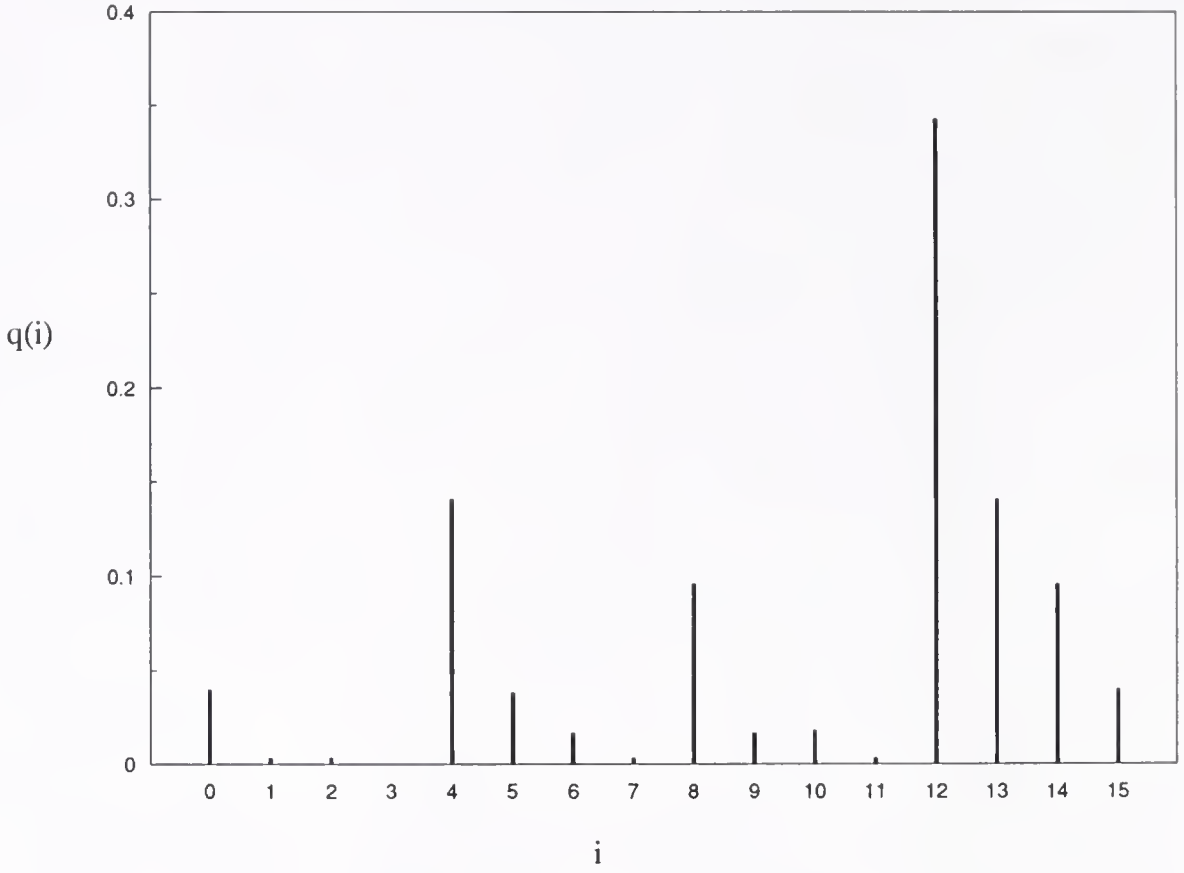
Limiting Stationary Distribution at M=3, L=4

Figure 5-8



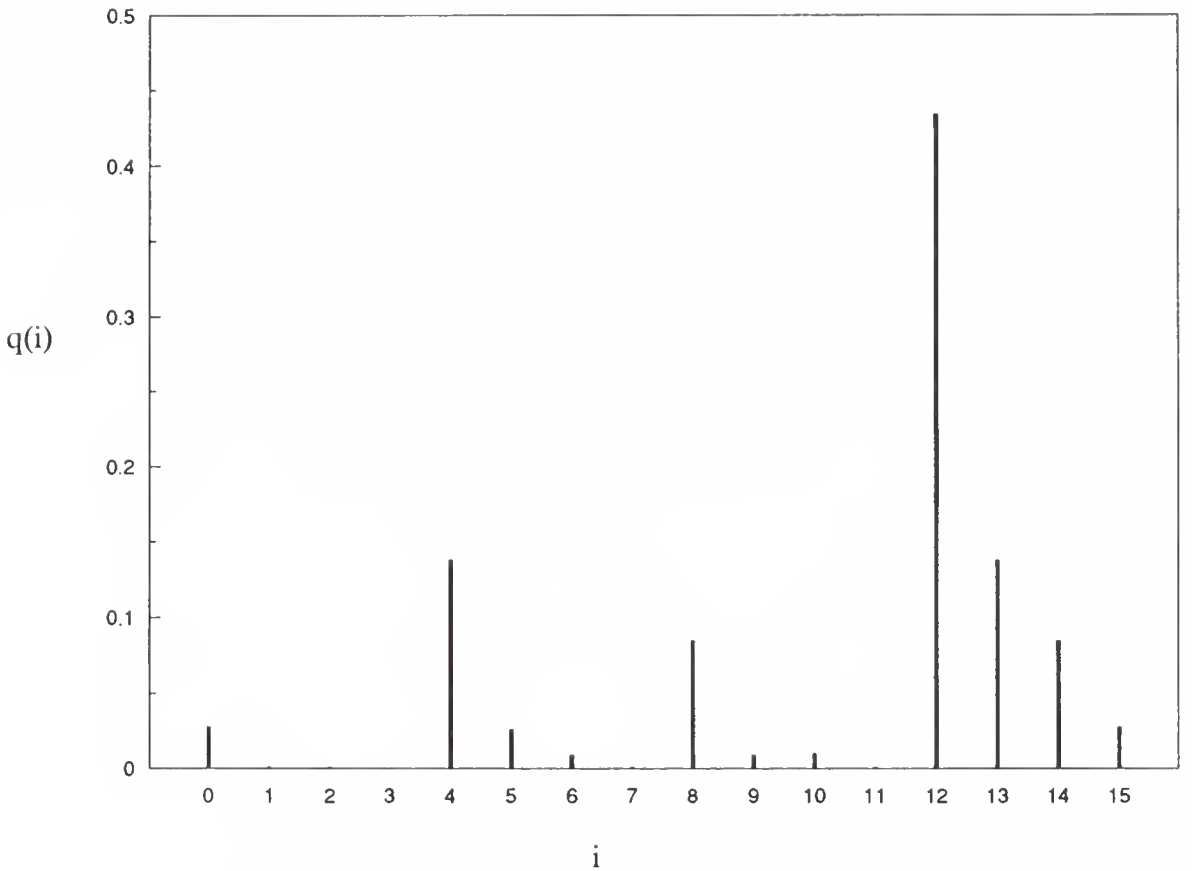
Limiting Stationary Distribution at M=4, L=4

Figure 5-9



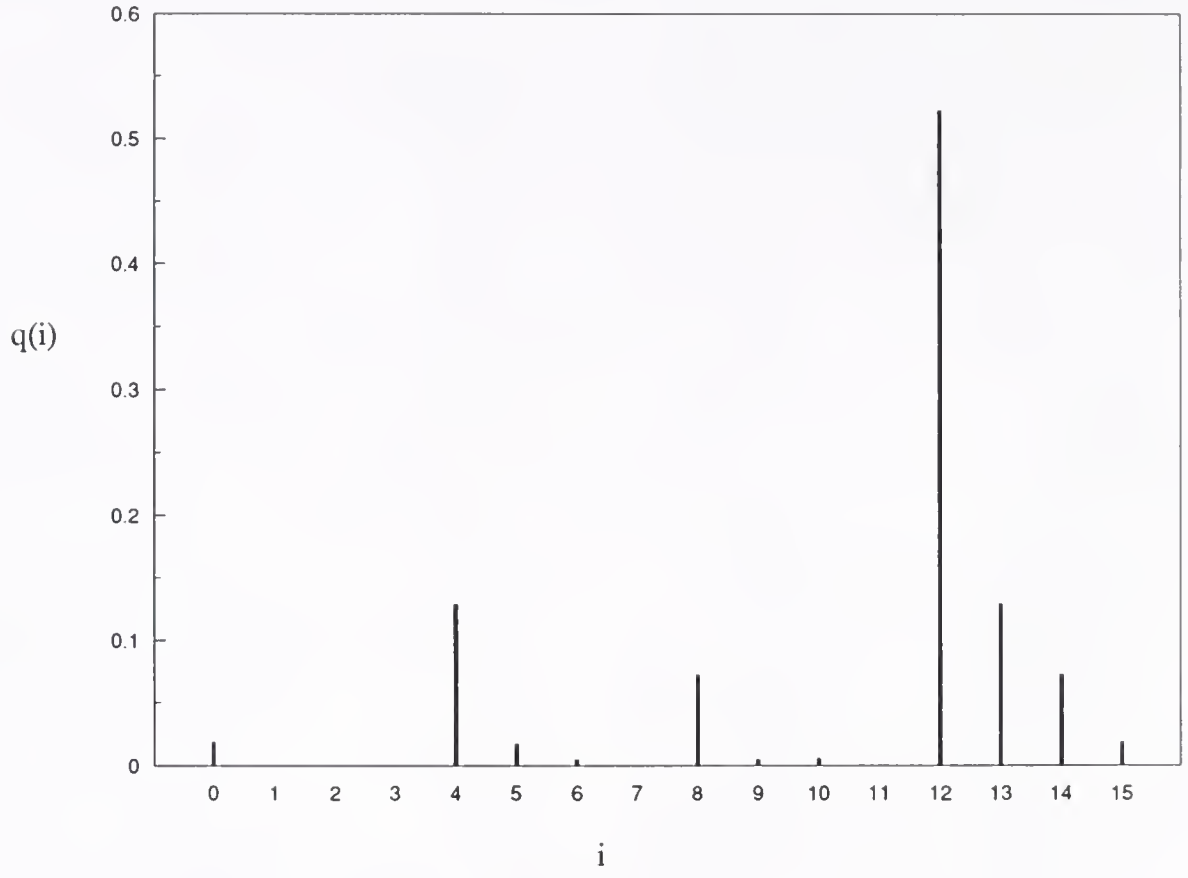
Limiting Stationary Distribution at M=5, L=4

Figure 5-10



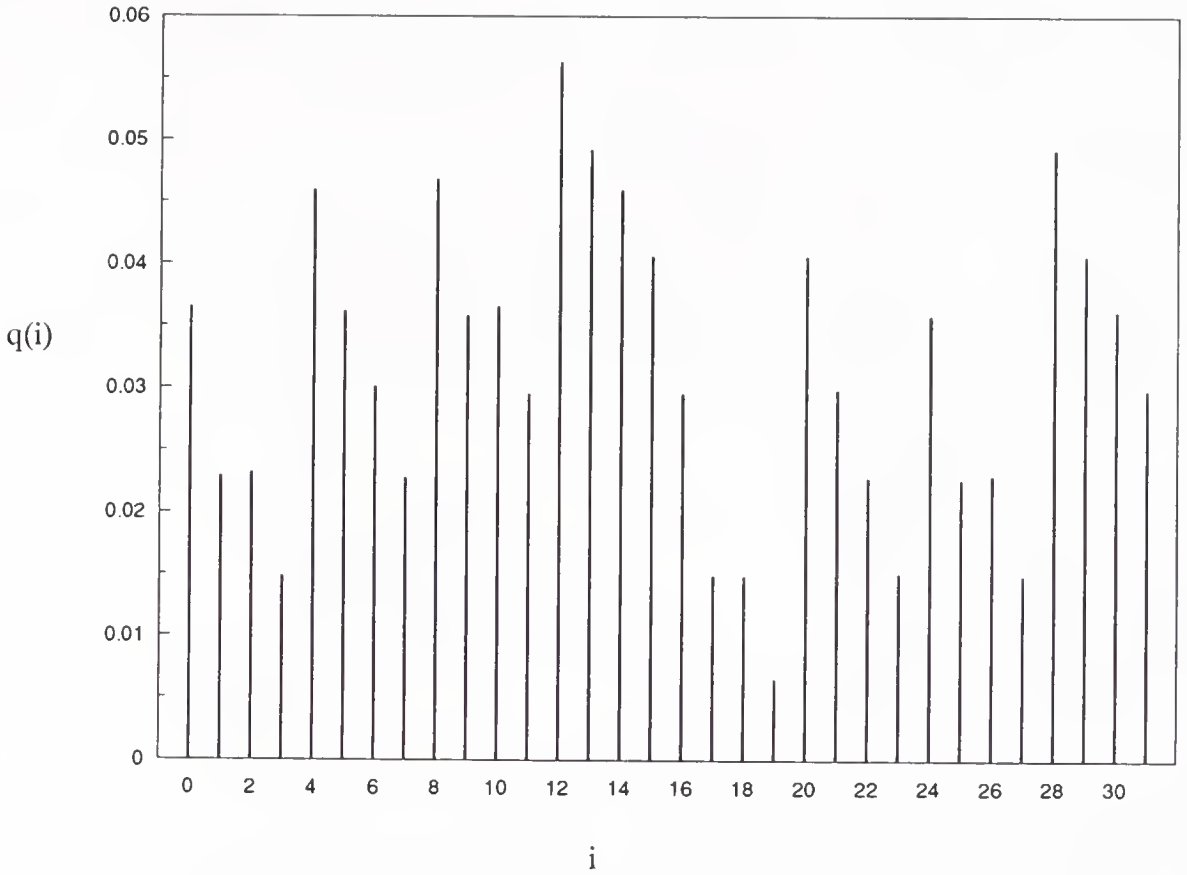
Limiting Stationary Distribution at M=6, L=4

Figure 5-11



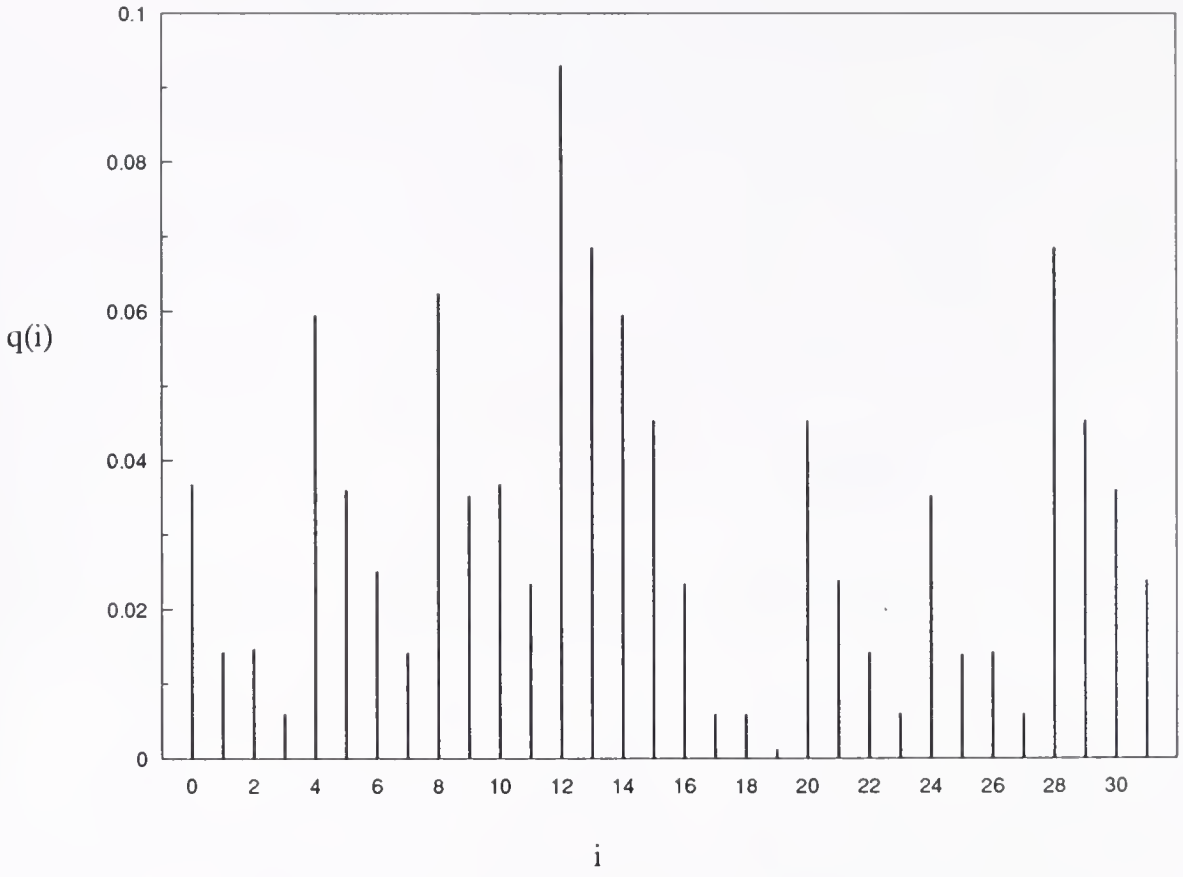
Limiting Stationary Distribution at M=7, L=4

Figure 5-12



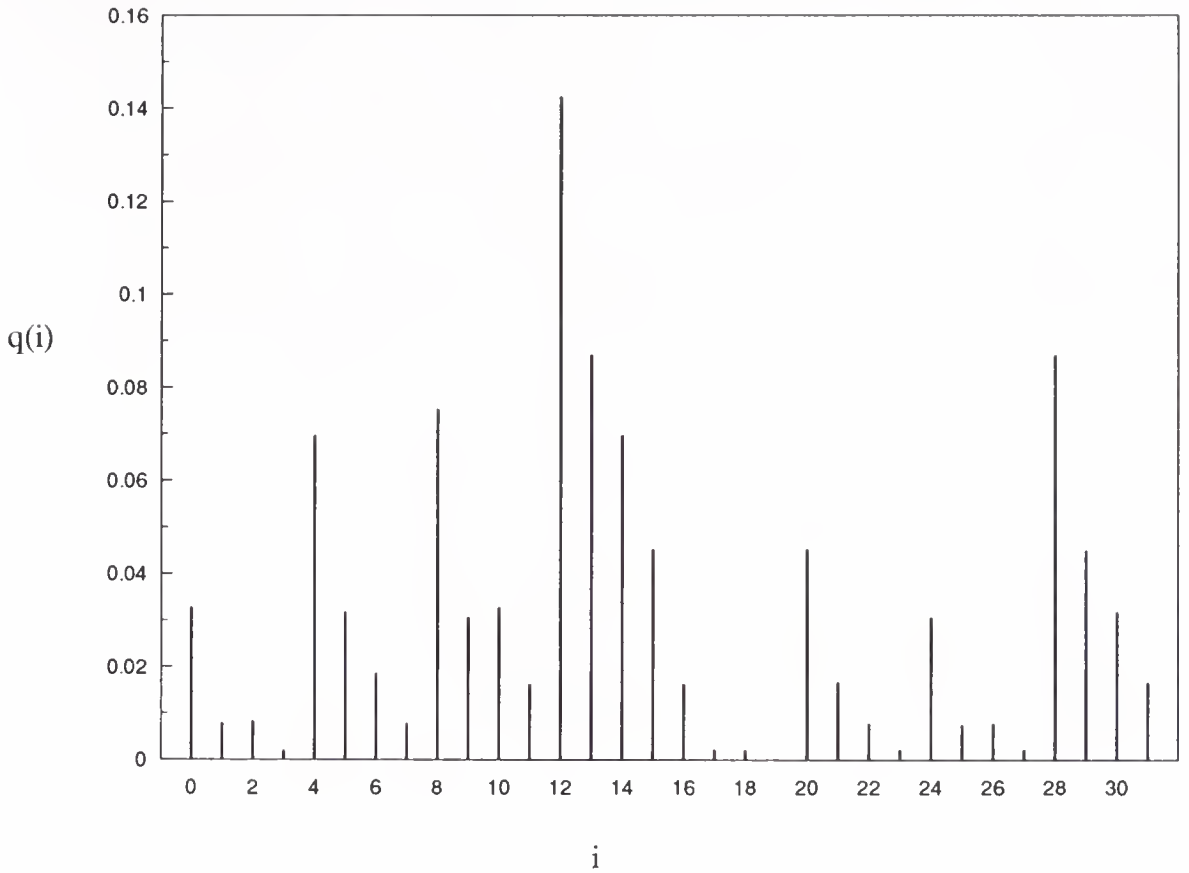
Limiting Stationary Distribution at M=2, L=5

Figure 5-13



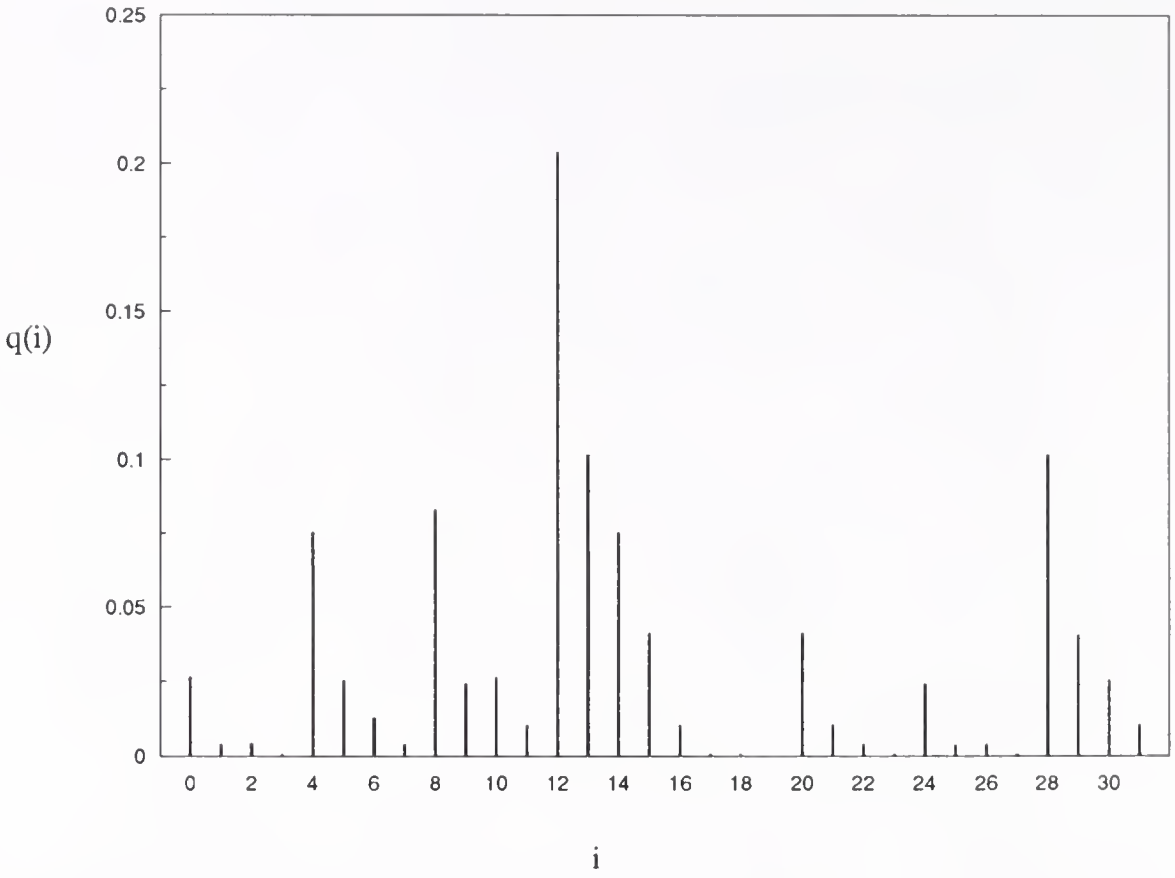
Limiting Stationary Distribution at M=3, L=5

Figure 5-14



Limiting Stationary Distribution at M=4, L=5

Figure 5-15



Limiting Stationary Distribution at M=5, L=5

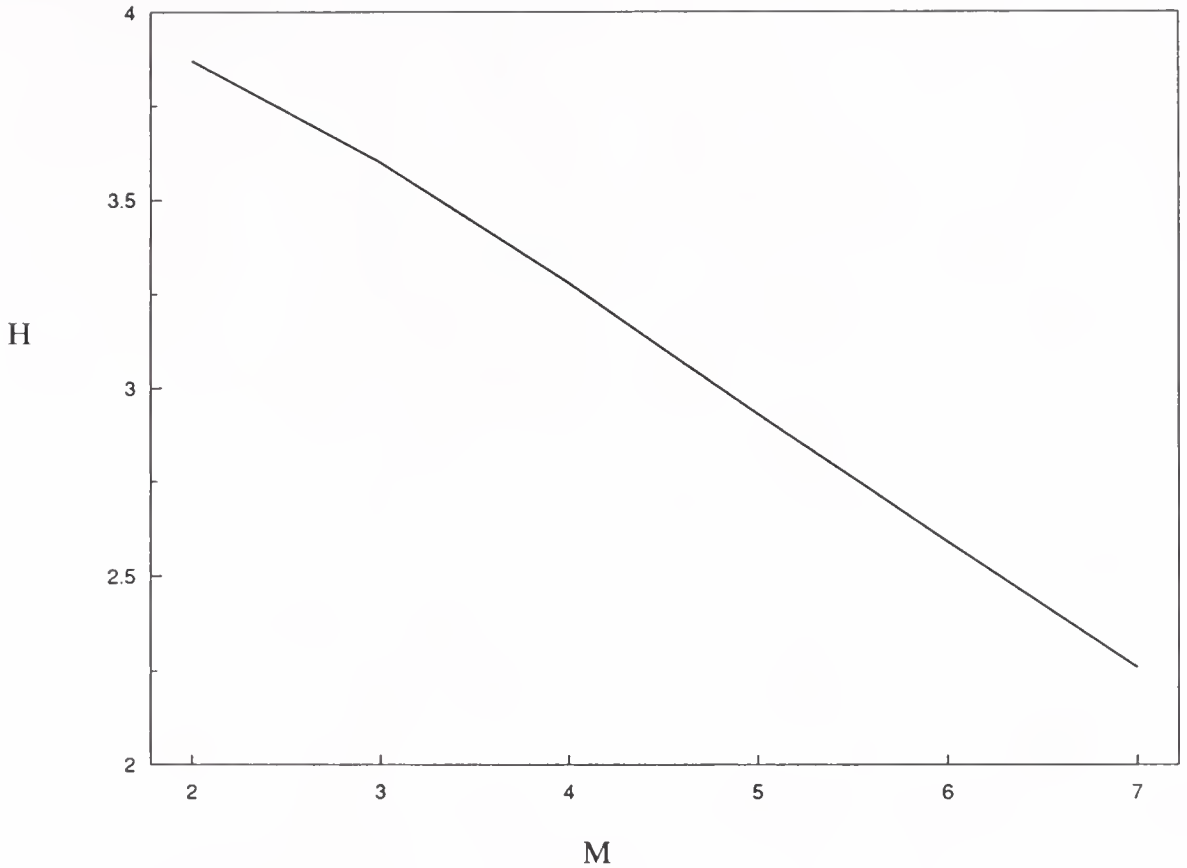
Figure 5-16

Table 5-5

CPU Utilization Statistics

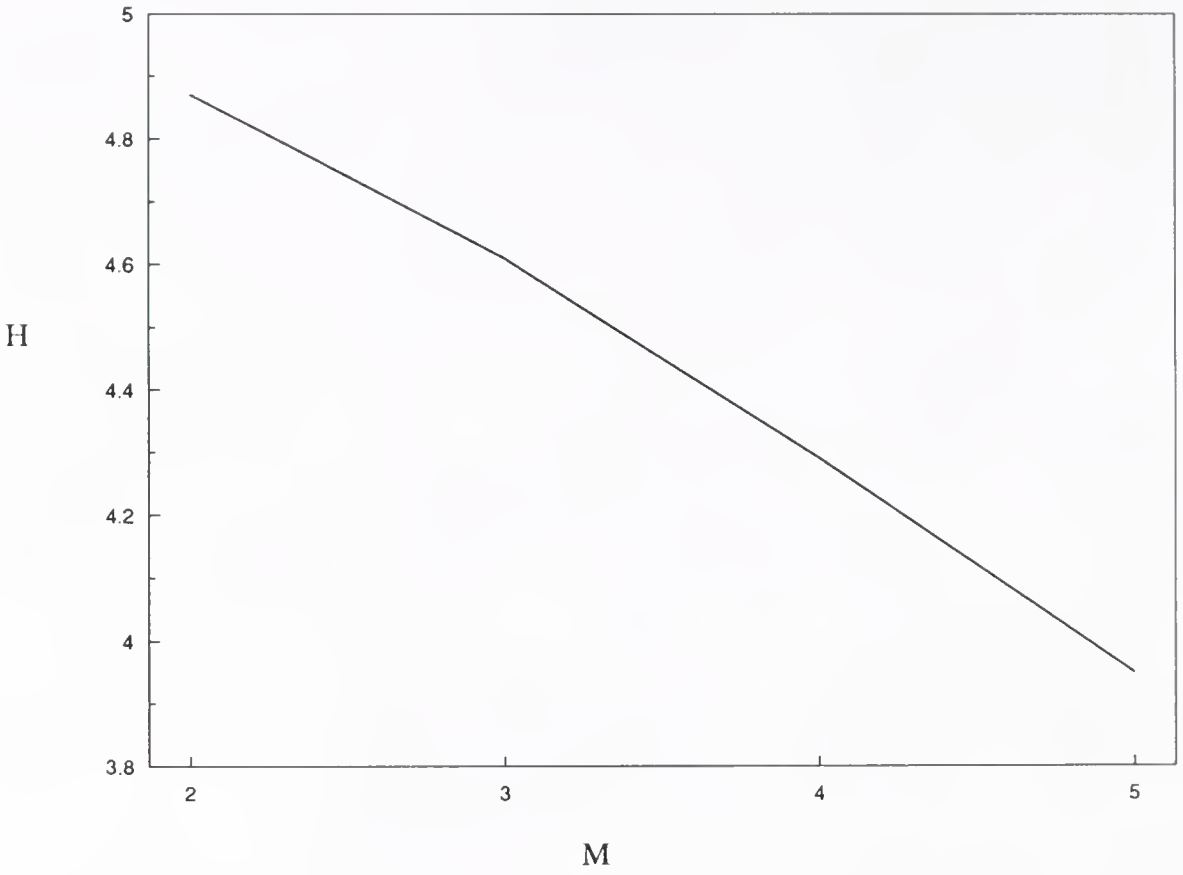
M	L	N'	Iterations	Seconds
2	4	136	8	<1
3	4	816	14	8
4	4	3876	19	86
5	4	15504	23	1219
6	4	54264	27	19048
7	4	170554	30	219930
2	5	528	9	13
3	5	5984	15	301
4	5	52360	21	11676
5	5	376992	26	**

** Not obtained due to unrecoverable log file error



Limiting Distribution Entropy vs Population Size (Four-Bit Problem)

Figure 5-17



Limiting Distribution Entropy vs Population Size (Five-Bit Problem)

Figure 5-18

SECTION 6

THE CRAMER'S RULE FORMULATION OF THE STATIONARY DISTRIBUTION

6.1 Overview

In Sections 4.3.2 and 4.3.3, the time-homogeneous two and three-operator simple genetic algorithm Markov chains are shown to possess unique stationary distributions. Those conclusions are established by invoking Theorem A3, which asserts that in each case the stationary distribution is a left eigenvector of the state transition matrix and that the additional constraint that it be a probability vector (Definition A2) makes the solution unique. In this section, the existence and uniqueness arguments are refined into a Cramer's rule formulation of the solution. This development concerns the time-homogeneous algorithms only, with α constrained to $\alpha > 0$, and it appeals heavily to the foundation provided in Appendix B.

The product of this development is an expression for the components of the stationary distribution vector as rational functions generated from the characteristic polynomials of matrices derived from the state transition matrix. The derived matrices are generated by setting selected rows of $\overline{P}$ to zero. The utility of the approach is that the form of $\overline{P}$ suggests a mechanism for expressing the values of the characteristic polynomials. Some key intermediate parts of the required methodology are developed in Section 9, but the effort stops short of explicit solution. However, some very significant conclusions concerning the asymptotic behavior of the algorithm are obtainable (Sections 7 and 8) from the results developed here without explicitly solving the system.

6.2 The Stationary Distribution Description

As established in Section 4, implementation of the mutation operator with nonzero mutation probability (i.e. $\alpha > 0$) implies that for both the two and three-operator

algorithms, $\forall \overline{m}, \overline{n} \in S' : P(\overline{m} \mid \overline{n}) > 0$. Thus, by Definition B1, $\overline{P}$ is primitive for any integer $k \geq 1$. Hence, from Section B.3, the stationary distribution of the two and three-operator simple genetic algorithm exists, is unique and is a left eigenvector of the state transition matrix corresponding to eigenvalue 1, i.e.

$$\overline{q}_\alpha^T \overline{P} = \overline{q}_\alpha^T$$

or equivalently

$$\overline{q}_\alpha^T (\overline{P} - \overline{I}) = \overline{0}^T. \qquad \text{Eq. 6.1}$$

The following proposition establishes a significant fact concerning the rank of the matrix $(\overline{P} - \overline{I})$ in Eq. 6.1.

Proposition 6.1: The rank of the matrix $(\overline{P} - \overline{I})$ in Eq. 6.1 is exactly $N' - 1$ where $N' = \text{card}(S')$ is the dimension of $\overline{P}$.

This result follows from Theorem B4(f). Its significance is that exactly one column of the system of equations in Eq. 6.1 can be replaced without sacrificing any of the constraints which Eq. 6.1 imposes on $\overline{q}_\alpha$. Proposition 6.2 below concerns such a modification of the system. The modification consists of replacing any column (e.g. the column indexed by $\overline{n} \in S'$) of Eq. 6.1 by a column corresponding to the constraint

$$\sum_{\overline{m} \in S'} q_\alpha(\overline{m}) = \overline{q}_\alpha^T \overline{1} = 1, \qquad \text{Eq. 6.2}$$

thus producing a system of the form

$$\overline{q}_\alpha^T (\overline{P} - \overline{I})_{\overline{n}} = \overline{e}_{\overline{n}}^T \qquad \text{Eq. 6.3}$$

where $(\overline{P} - \overline{I})_{\overline{n}}$ is generated by replacing the column of $(\overline{P} - \overline{I})$ indexed by $\overline{n} \in S'$ with the vector $\overline{1}$ whose components all have the value 1, and where $\overline{e}_{\overline{n}}^T$ is the row vector containing 1 in column $\overline{n}$ and 0's elsewhere.

Proposition 6.2: If the constraint described in Eq. 6.2 is used to replace any column (e.g. column $\bar{n}$) of the system in Eq. 6.1, the resulting system (Eq. 6.3) is full rank, or equivalently, $|(\bar{P}-\bar{I})_{\bar{n}}| \neq 0$.

Since $\bar{P}$ is a stochastic matrix (Definition A3), the system of equations in Eq 6.1 can be transformed into an equivalent system in which the column indexed by the arbitrary column index $\bar{n} \in S'$ is represented by the equation $\vec{q}_\alpha^{\,T}\bar{0} = 0$. The required transformation is obtainable by replacing column $\bar{n}$ by the sum of all columns $\bar{m} \in S'$, and thus any $\bar{n} \in S'$ is a candidate for replacement. Proposition 6.2 is then a restatement of Proposition B2 in terms of the determinant of the matrix of the modified system. It is the essential condition for justification of the following proposition.

Proposition 6.3: The components of the stationary distribution can be expressed in the form

$$q_\alpha(\bar{m}) = \frac{|(\bar{P}-\bar{I})_{\bar{n}}^{(\bar{m})}|}{|(\bar{P}-\bar{I})_{\bar{n}}|}$$

where $(\bar{P}-\bar{I})_{\bar{n}}^{(\bar{m})}$ is derived from $(\bar{P}-\bar{I})_{\bar{n}}$ by replacing the row of $(\bar{P}-\bar{I})_{\bar{n}}$ indexed by $\bar{m} \in S'$ with the row vector $\vec{e}_{\bar{n}}^{\,T}$.

This result is simply an application of Cramer's Rule to the solution of the system in Eq. 6.3. It applies because $|(\bar{P}-\bar{I})_{\bar{n}}| \neq 0$ is assured by Proposition 6.2.

The equality defined in Proposition 6.3 can be evaluated without computing $|(\bar{P}-\bar{I})_{\bar{n}}|$ directly, as suggested by the following proposition.

Proposition 6.4: The denominator determinant in Proposition 6.3 can be written as

$$|(\bar{P}-\bar{I})_{\bar{n}}| = \sum_{\bar{m} \in S'} \left|(\bar{P}-\bar{I})_{\bar{n}}^{(\bar{m})}\right|.$$

This result follows from application of elementary column operations on column $\bar{n}$ of $|(\bar{P}-\bar{I})_{\bar{n}}|$ and employing the definition of $|(\bar{P}-\bar{I})_{\bar{n}}^{(\bar{m})}|$. The essential step is noting that the cofactor of each of the (unit) elements in column $\bar{n}$ is equal to the corresponding $|(\bar{P}-\bar{I})_{\bar{n}}^{(\bar{m})}|$.

Since the numerator determinant defined in Proposition 6.3 is generated from $(\bar{P}-\bar{I})_{\bar{n}}$ by replacement of row $\bar{m}$ by the row vector $\bar{e}_{\bar{n}}^{T}$, its value is the cofactor of the (unit) element in row $\bar{m}$ and column $\bar{n}$. As indicated in the following proposition, it is equal to the determinant which results from the corresponding row replacement in $(\bar{P}-\bar{I})$.

Proposition 6.5: The numerator determinant defined in Proposition 6.3 can be written as

$$\left|(\bar{P}-\bar{I})_{\bar{n}}^{(\bar{m})}\right| = |(\bar{P}-\bar{I})^{(\bar{m}(\bar{n}))}|$$

where $(\bar{P}-\bar{I})^{(\bar{m}(\bar{n}))}$ is defined as the matrix which results from $(\bar{P}-\bar{I})$ by replacing the row indexed by $\bar{m}$ with the row vector $\bar{e}_{\bar{n}}^{T}$.

Next, note that if $\bar{m}=\bar{n}$, then $|(\bar{P}-\bar{I})^{(\bar{m}(\bar{n}))}|$ can be written as

$$|(\bar{P}-\bar{I})^{(\bar{m}(\bar{m}))}| = -\left|\left(\bar{P}_{\bar{m}}^{(\bar{0})}-\bar{I}\right)\right|$$

where $\bar{P}_{\bar{m}}^{(\bar{0})}$ is defined as the matrix which results by replacing row $\bar{m}$ of $\bar{P}$ by the row vector $\bar{0}^{T}$. If $\bar{m}\neq\bar{n}$, then by writing the replacement row in $|(\bar{P}-\bar{I})^{(\bar{m}(\bar{n}))}|$ as

$$\bar{e}_{\bar{n}}^{T} = [\bar{e}_{\bar{n}}^{T}-\bar{e}_{\bar{m}}^{T}+\bar{e}_{\bar{m}}^{T}] = [\bar{e}_{\bar{n}}^{T}-\bar{e}_{\bar{m}}^{T}-(-\bar{e}_{\bar{m}}^{T})],$$

$|(\bar{P}-\bar{I})^{(\bar{m}(\bar{n}))}|$ can be written as the difference of two determinants derived from $(\bar{P}-\bar{I})$, one with the $\bar{m}^{th}$ row replaced by $[\bar{e}_{\bar{n}}^{T}-\bar{e}_{\bar{m}}^{T}]$ and the second with the $\bar{m}^{th}$ row replaced by $[-\bar{e}_{\bar{m}}^{T}]$. The $-\bar{e}_{\bar{m}}^{T}$ term in each row replacement provides the necessary principal diagonal contribution to permit expression of $|(\bar{P}-\bar{I})^{(\bar{m}(\bar{n}))}|$ as

$$|(\bar{P}-\bar{I})^{(\bar{m}(\bar{n}))}| = \left|\left(\bar{P}_{\bar{m}}^{(\bar{n})}-\bar{I}\right)\right| - \left|\left(\bar{P}_{\bar{m}}^{(\bar{0})}-\bar{I}\right)\right|$$

where $\bar{P}_{\bar{m}}^{(\bar{0})}$ is defined as before and where $\bar{P}_{\bar{m}}^{(\bar{n})}$ is defined as the matrix which results by replacing row $\bar{m}$ of $\bar{P}$ by the row vector $\bar{e}_{\bar{n}}^{T}$.

This result can be further reduced by noting that the row replacement by which $\bar{P}_{\bar{m}}^{(\bar{n})}$ is generated from $\bar{P}$ preserves the row sum constraint (i.e. $\bar{P}_{\bar{m}}^{(\bar{n})}$ is a stochastic matrix). Thus, 1 is an eigenvalue of $\bar{P}_{\bar{m}}^{(\bar{n})}$ (Definition A3), from which it follows that $|(\bar{P}_{\bar{m}}^{(\bar{n})} - \bar{I})| = 0$. Consequently, the $\bar{m} = \bar{n}$ and $\bar{m} \neq \bar{n}$ cases can be assembled as indicated by the following proposition.

Proposition 6.6: The determinant $|(\bar{P} - \bar{I})^{(\bar{m}(\bar{n}))}|$ defined in Proposition 6.5 can be written as

$$|(\bar{P} - \bar{I})^{(\bar{m}(\bar{n}))}| = -\left|\left(\bar{P}_{\bar{m}}^{(\bar{0})} - \bar{I}\right)\right|.$$

By collecting the results of Propositions 6.3-6 and noting that the superscript in $\bar{P}_{\bar{m}}^{(\bar{0})}$ is now superfluous, the components of the stationary distribution can be written as indicated in the following proposition.

Proposition 6.7: The components of the stationary distribution can be expressed in the form

$$q_{\alpha}(\bar{m}) = \frac{-|\bar{P}_{\bar{m}} - \bar{I}|}{\sum_{\bar{n} \in S'} -|\bar{P}_{\bar{n}} - \bar{I}|} = \frac{|\bar{P}_{\bar{m}} - \bar{I}|}{\sum_{\bar{n} \in S'} |\bar{P}_{\bar{n}} - \bar{I}|}$$

where $\bar{P}_{\bar{m}}$ and $\bar{P}_{\bar{n}}$ are derived from $\bar{P}$ by replacing the rows indexed by $\bar{m}$ and $\bar{n}$ respectively with the row vector $\bar{0}^{T}$.

Thus, computing the stationary distribution components reduces to evaluating the characteristic polynomials of the $\bar{P}_{\bar{m}}$'s at $\lambda = 1$ (i.e. $\phi_{\bar{m}}(\lambda) = |\bar{P}_{\bar{m}} - \lambda\bar{I}| \Rightarrow |\bar{P}_{\bar{m}} - \bar{I}| = \phi_{\bar{m}}(1)$). Also, since 1 is an eigenvalue of $\bar{P}$ it follows that $\phi(1) = |\bar{P} - \bar{I}| = 0$, which suggests the following alternative to Proposition 6.7. Its usefulness is established in Sections 9.3-9.4.

Proposition 6.8: The components of the stationary distribution can be expressed in the alternative form

$$q_\alpha(\overline{m}) = \frac{|\overline{P} - \overline{I}| - |\overline{P}_{\overline{m}} - \overline{I}|}{\sum_{\overline{n} \in S'} (|\overline{P} - \overline{I}| - |\overline{P}_{\overline{n}} - \overline{I}|)}$$

where as before $\overline{P}_{\overline{m}}$ and $\overline{P}_{\overline{n}}$ are derived from $\overline{P}$ by replacing the rows indexed by $\overline{m}$ and $\overline{n}$ respectively with the row vector $\overline{0}^T$.

6.3 Positivity of the Stationary Distribution Components

Strict positivity of the stationary distribution components can be deduced from Theorem B4 and the form of $\overline{P}_{\overline{m}}$. Every element of $\overline{P}_{\overline{m}}$ in every row other than row $\overline{m}$ is identical to the corresponding element of $\overline{P}$, while those in row $\overline{m}$ are zero. This is expressed in the nonnegative matrix notation of Appendix B as $0 \le \overline{P}_{\overline{m}} \le \overline{P}$. Further, since $\overline{P}_{\overline{m}}$ and $\overline{P}$ differ in row $\overline{m}$, $\overline{P}_{\overline{m}} \neq \overline{P}$, and consequently by Theorem B4(e), every eigenvalue of $\overline{P}_{\overline{m}}$ satisfies $|\lambda_i| < 1$. It follows that for $\lambda \ge 1$, (1) $|\overline{P}_{\overline{m}} - \lambda\overline{I}| = \prod(\lambda_i - \lambda) \neq 0$ and (2) the algebraic sign of $|\overline{P}_{\overline{m}} - \lambda\overline{I}|$ is $(-1)^{N'}$ for all $\overline{m} \in S'$. Specializing these arguments to the case $\lambda = 1$ yields the following proposition.

Proposition 6.9: For all $\alpha > 0$, the value of the determinant $|\overline{P}_{\overline{m}} - \overline{I}|$ satisfies

$\forall \overline{m} \in S'$: (1) $|\overline{P}_{\overline{m}} - \overline{I}| \neq 0$ and

(2) the algebraic sign of $|\overline{P}_{\overline{m}} - \overline{I}|$ is $(-1)^{N'}$.

An immediate consequence of Proposition 6.9 is that both numerator and denominator of the expression for $\overline{q}_\alpha(\overline{m})$ in Proposition 6.7 are nonzero and have identical algebraic sign. Strict positivity of the stationary distribution components follows from these observations. That is, $\forall \overline{m} \in S' : q_\alpha(\overline{m}) > 0$.

6.4 The Indeterminate Form at $\alpha = 0$

All of the results established in this section assume that the mutation probability parameter is strictly positive ($\alpha > 0$), and thus are not applicable at $\alpha = 0$. The reason is apparent when Eq. 4.7 and the two-operator result in 4.17 (or the three-operator counterparts of Eq. 4.17 given by Eq. 4.23 and 4.27) are applied to $|\overline{P}_{\overline{m}} - \overline{I}|$. It follows that the row of the $\alpha \to 0^+$ limit of $|\overline{P}_{\overline{m}} - \overline{I}|$ corresponding to the one-operator absorbing state $\overline{n}_A \in S_A', \overline{n}_A \neq \overline{m}$ is zero. That is, the only nonzero entry in row $\overline{n}_A$ of the $\alpha \to 0^+$ limit of $\overline{P}_{\overline{m}}$ is the principal diagonal element

$$\lim_{\alpha \to 0^+} P_2(\overline{n}_A \mid \overline{n}_A) = \lim_{\alpha \to 0^+} P_3(\overline{n}_A \mid \overline{n}_A) = P_1(\overline{n}_A \mid \overline{n}_A) = 1,$$

which is cancelled by the corresponding principal diagonal element in $-\overline{I}$. Thus,

$$\forall \overline{m} \in S' : \lim_{\alpha \to 0^+} |\overline{P}_{\overline{m}} - \overline{I}| = 0,$$

and consequently Propositions 6.7 and 6.8 yield indeterminate forms. However, as demonstrated in the following section, it is possible to verify that a limiting stationary distribution vector exists for the time-homogeneous two and three-operator algorithms.

SECTION 7

THE ZERO MUTATION PROBABILITY STATIONARY DISTRIBUTION LIMIT

7.1 Overview

In Section 4.3.1, it is established that the time-homogeneous one-operator genetic algorithm Markov chain possesses a stationary distribution but that it is not unique. In Sections 4.3.2 and 4.3.3, it is established that the time-homogeneous two and three-operator counterparts possess unique stationary distributions provided $\alpha > 0$, and Section 6 formulates the existence and uniqueness argument into a rational function expression for the unique solution. Since the two-operator state transition matrix approaches its one-operator counterpart as $\alpha \rightarrow 0^+$ (Eq. 4.17) and since the three-operator algorithm exhibits the corresponding behavior with respect to the $P_2'(i \mid \overline{n})$s (Eq. 4.23), a question which naturally arises from these observations is whether an $\alpha \rightarrow 0^+$ limiting distribution exists for the two and three-operator algorithms. (If such a limit exists, then it is necessarily unique). This section answers that question affirmatively and also confirms the observation made in Section 5.5 that the limiting distribution is nonzero for all states corresponding to uniform populations (absorbing states).

The approach taken here is to transform the expressions for $q_\alpha(\overline{m})$ in Propositions 6.7 and 6.8 into equivalent expressions which yield determinate forms at $\alpha = 0$. The result requires transforming $\overline{P}$ and $\overline{P}_{\overline{m}}$ into related matrices but with the states corresponding to uniform populations (one-operator absorbing states) coalesced into adjacent nonuniform population states. The development is tedious and involves some additional notation.

7.2 Functional Form of the Stationary Distribution

Before proceeding with the limiting case development which is the primary purpose of this section, it is convenient to establish some intermediate results concerning the

behavior of $\bar{q}_\alpha$ as a function of α. These results follow from the results developed in Section 6 and some simple observations about the form of the elements of $\bar{\bar{P}}$.

From Eq. 4.14-4.16 and Eq. 4.22-26, all elements of the state transition matrix are rational functions of α with denominator polynomial $(1+\alpha)^{ML}$. Thus, for $\alpha > 0$

$$(1+\alpha)^{ML\cdot N'} |\bar{P}_{\bar{m}} - \bar{I}| = |(1+\alpha)^{ML}\bar{P}_{\bar{m}} - (1+\alpha)^{ML}\bar{I}|$$

$$= |\bar{Q}_{\bar{m}} - (1+\alpha)^{ML}\bar{I}|$$

where every element of $\bar{Q}_{\bar{m}}$ (and hence the value of $|\bar{Q}_{\bar{m}} - (1+\alpha)^{ML}\bar{I}|$) is a polynomial in α. Further, since row $\bar{m}$ in $\bar{Q}_{\bar{m}}$ is zero, the polynomial value of the determinant includes the factor $(1+\alpha)^{ML}$. Consequently

$$(1+\alpha)^{ML(N'-1)} |\bar{P}_{\bar{m}} - \bar{I}| = \theta_{\bar{m}}(\alpha) \qquad \text{Eq. 7.1}$$

for $\theta_{\bar{m}}(\alpha)$ some polynomial function of α. Proposition 7.1 below follows.

Proposition 7.1: For all $\alpha > 0$, the value of the determinant $|\bar{P}_{\bar{m}} - \bar{I}|$ is a rational function of α with nonzero denominator polynomial $(1+\alpha)^{ML(N'-1)}$.

By applying Eq. 7.1 to Proposition 6.7, the components of $\bar{q}_\alpha$ can be written as

$$q_\alpha(\bar{m}) = \frac{\theta_{\bar{m}}(\alpha)}{\sum_{\bar{n} \in S'} \theta_{\bar{n}}(\alpha)} = \frac{\theta_{\bar{m}}(\alpha)}{\Theta(\alpha)}. \qquad \text{Eq. 7.2}$$

Hence, the $q_\alpha(\bar{m})$ are rational functions of α, and since a rational function is continuous everywhere its denominator polynomial is nonzero, application of Proposition 6.9 and Eq. 7.1 (which together establish that $\Theta(\alpha) = \sum \theta_{\bar{n}}(\alpha) \neq 0$) to Eq. 7.2 yields the following.

Proposition 7.2: For all $\alpha > 0$, the components of $\bar{q}_\alpha$ are continuous rational functions of the independent variable α.

Further, differentiation of Eq. 7.2 with respect to alpha yields a rational function of α

$$\frac{dq_\alpha(\bar{m})}{d\alpha} = \frac{1}{\Theta(\alpha)^2}\left[\Theta(\alpha)\frac{d\theta_{\bar{m}}(\alpha)}{d\alpha} - \theta_{\bar{m}}(\alpha)\frac{d\Theta(\alpha)}{d\alpha}\right] \qquad \text{Eq. 7.3}$$

with nonzero denominator polynomial $\Theta(\alpha)^2$. The following proposition is a consequence.

Proposition 7.3: For all $\alpha > 0$, the components of the first derivative of $\bar{q}_\alpha$ with respect to α are continuous rational functions of α.

7.3 The Absorbing State Rows of $|\bar{P} - \bar{I}|$ and $|\bar{P}_{\bar{m}} - \bar{I}|$

The rows corresponding to one-operator absorbing states in the determinant $|\bar{P} - \bar{I}|$ have a particularly simple form. The nondiagonal elements of row $\bar{n}_A \in S_A'$, which represents a uniform population of solutions $i_A \in S$, are given by Eq. 4.19 and 4.29 respectively for the two and three-operator cases. The principal diagonal element is obtained by evaluating Eq. 4.19 or 4.29 at $\bar{m} = \bar{n}_A$ and subtracting 1. Thus,

$$\begin{aligned} P(\bar{n}_A \mid \bar{n}_A) - 1 &= \binom{M}{\bar{n}_A}\frac{\alpha^{MH(i_A, i_A)}}{(1+\alpha)^{ML}} - 1 \\ &= \frac{1}{(1+\alpha)^{ML}} - 1 \\ &= \frac{1 - 1 - ML\alpha - ML(ML-1)\alpha^2/2 - \cdots - \alpha^{ML}}{(1+\alpha)^{ML}} \\ &= \frac{-ML\alpha + O(\alpha^2)}{(1+\alpha)^{ML}}, \end{aligned}$$

and if the general element in $|\bar{P} - \bar{I}|$ is denoted by $T(\bar{m} \mid \bar{n})$, then the elements of row $\bar{n}_A$ can be written as

$$\forall \bar{n}_A \in S_A' : T(\bar{n} \mid \bar{n}_A) = \begin{cases} \dfrac{-ML\alpha + O(\alpha^2)}{(1+\alpha)^{ML}} & \bar{n} = \bar{n}_A \\ \dbinom{M}{\bar{n}}\dfrac{\alpha^{\sum n(i) I(i, i_A)}}{(1+\alpha)^{ML}} & \bar{n} \in S' - \{\bar{n}_A\} \end{cases} . \qquad \text{Eq. 7.4}$$

Additional insight into the form of the absorbing state rows can be obtained with the aid of the following notation. Let $\bar{m}_A, \bar{n}_A \in S_A'$ be distinct but otherwise arbitrary absorbing states of the one-operator Markov chain, let $i_A \in S$ be the bit-string represented in $\bar{n}_A$ and let $S(i_A) \subset S$ be the set of bit-strings accessible from i_A via exactly one bit mutation event (i.e. $S(i_A) = \{i_1 : i_1 \in S, H(i_1, i_A) = 1\}$). It follows from this definition that $\text{card}(S(i_A)) = L$. Then, for $M > 1$ let $S(\bar{n}_A)'$ be defined as

$$S(\bar{n}_A)' = \{\bar{n} : \bar{n} \in S', n(i_A) = M - 1, n(i_1) = 1, i_1 \in S(i_A)\} \subset S',$$

the set of nonabsorbing states adjacent to the absorbing state $\bar{n}_A$. The restriction on M is required to ensure that no absorbing state $\bar{m}_A$ is contained in the adjacency set of any absorbing state $\bar{n}_A$. $S(\bar{n}_A)'$ includes exactly one distinct element for each $i_1 \in S(i_A)$, and consequently $\text{card}(S(\bar{n}_A)') = \text{card}(S(i_A)) = L$. Also, from the form of $S(\bar{n}_A)'$, it follows that for $M \geq 3$, $S(\bar{m}_A)'$ and $S(\bar{n}_A)'$ are disjoint if $\bar{m}_A$ and $\bar{n}_A$ are distinct one-operator absorbing states. Thus, if S_A'' is defined as

$$S_A'' = \bigcup_{\bar{n}_A \in S_A'} S(\bar{n}_A)'$$

and $M \geq 3$, then $\text{card}(S_A'') = \text{card}(S_A') \times L = NL$. This restriction on M is assumed in all of the following.

With the aid of the new notation, the element in column $\bar{n} \in S(\bar{n}_A)'$ of row $\bar{n}_A$ in $|\bar{P} - \bar{I}|$ can be written as

$$\begin{aligned} \forall \bar{n} \in S(\bar{n}_A)' : T(\bar{n} \mid \bar{n}_A) = P(\bar{n} \mid \bar{n}_A) &= \binom{M}{M-1} \frac{\alpha}{(1+\alpha)^{ML}} \\ &= \binom{M}{1} \frac{\alpha}{(1+\alpha)^{ML}} \\ &= \frac{M\alpha}{(1+\alpha)^{ML}}. \end{aligned}$$

Thus, Eq. 7.4 can be revised as follows

$$\forall \bar{n}_A \in S_A' : T(\bar{n} \mid \bar{n}_A) = \begin{cases} \dfrac{-ML\alpha + O(\alpha^2)}{(1+\alpha)^{ML}} & \bar{n} = \bar{n}_A \\ \dfrac{M\alpha}{(1+\alpha)^{ML}} & \bar{n} \in S(\bar{n}_A)' \\ \dfrac{O(\alpha^s)}{(1+\alpha)^{ML}} & \bar{n} \in S' - S(\bar{n}_A)' - \{\bar{n}_A\} \end{cases} \qquad \text{Eq. 7.5}$$

where the exponent s of α in the order expression for the general term is an integer satisfying $s \geq 2$. The elements in columns $\bar{n}_A$ and $\bar{n} \in S(\bar{n}_A)'$ are first order in α while the elements in all other columns are at least second order.

Eq. 7.5 applies to every absorbing state row of $|\bar{P}_{\bar{m}} - \bar{I}|$ as well if $\bar{m} \notin S_A'$. If $\left|\bar{P}_{\bar{m}_A} - \bar{I}\right|$ is being considered where $\bar{m}_A \in S_A'$, then row $\bar{m}_A$ contains -1 at its principal diagonal and zeros elsewhere. In that case Eq. 7.5 only applies to the absorbing state rows $\bar{n}_A \in S_A' - \{\bar{m}_A\}$. Exactly $N-1$ such rows exist in $\left|\bar{P}_{\bar{m}_A} - \bar{I}\right|$.

By applying Eq. 7.5 and these observations to Proposition 7.1, it follows that the lowest order term with nonzero coefficient which can conceivably exist in the numerator polynomial of $\left|\bar{P}_{\bar{m}_A} - \bar{I}\right|$ is the order α^{N-1} term. Similar reasoning reveals that the corresponding lowest order term with nonzero coefficient for $|\bar{P}_{\bar{m}} - \bar{I}|$ with $\bar{m} \notin S_A'$ is the order α^N term. If the coefficient of the order α^{N-1} term in the numerator polynomial of $\left|\bar{P}_{\bar{m}_A} - \bar{I}\right|$ is indeed nonzero, and if the corresponding coefficients for all such $\bar{m}_A$ have the same algebraic sign, then the required limiting value of $\bar{q}_\alpha$ can be expressed in terms of these nonzero coefficients via substitution into Proposition 6.7. These conditions are in fact satisfied as demonstrated below.

7.4 Reformulation of Propositions 6.7 and 6.8

The next step in this development is the definition of some auxiliary matrices related to $\bar{P}$ and $\bar{P}_{\bar{m}_A}$ and the reformulation of Propositions 6.7 and 6.8 in terms of them. The new matrices, designated $\bar{P}(\bar{m}_A)'$ and $\bar{P}_{\bar{m}_A}'$ respectively, are derived by coalescing each of the $N-1$ absorbing state columns $\bar{n}_A \in S_A' - \{\bar{m}_A\}$ of $|\bar{P} - \bar{I}|$ and $\left|\bar{P}_{\bar{m}_A} - \bar{I}\right|$ with its neighboring nonabsorbing state columns, $\bar{n} \in S(\bar{n}_A)'$. Specifically, let $\left|\bar{Q}_{\bar{m}_A}\right|$ be derived

from $\left|\bar{P}_{\bar{m}_A} - \bar{I}\right|$ by adding 1/L times the column $\bar{n}_A \in S_A' - \{\bar{m}_A\}$ to each of the L adjacent nonabsorbing columns $\bar{n} \in S(\bar{n}_A)'$ and repeating the process for each remaining $\bar{n}_A \in S_A' - \{\bar{m}_A\}$. This operation is applied once each for the exactly N − 1 absorbing state columns $\bar{n}_A \in S_A' - \{\bar{m}_A\}$ and it preserves the value of the determinant $\left|\bar{Q}_{\bar{m}_A}\right| = \left|\bar{P}_{\bar{m}_A} - \bar{I}\right|$.

If now $Q_{\bar{m}_A}(\bar{m} \mid \bar{n})$ denotes the general element of $\left|\bar{Q}_{\bar{m}_A}\right|$, then by applying the recipe used in its construction and Eq. 7.5, the elements in the absorbing state rows $\bar{n}_A \in S_A' - \{\bar{m}_A\}$ of $\left|\bar{Q}_{\bar{m}_A}\right|$ can be written as

$$\forall \bar{m}_A \in S_A', \forall \bar{n}_A \in S_A' - \{\bar{m}_A\} : Q_{\bar{m}_A}(\bar{m} \mid \bar{n}_A) = \begin{cases} \dfrac{-ML\alpha + O(\alpha^2)}{(1+\alpha)^{ML}} & \bar{m} = \bar{n}_A \\ \dfrac{O(\alpha^2)}{(1+\alpha)^{ML}} & \bar{m} \in S(\bar{n}_A)' \\ \dfrac{O(\alpha^s)}{(1+\alpha)^{ML}} & \bar{m} \in S' - S(\bar{n}_A)' - \{\bar{n}_A\} \end{cases}$$

where as before s is an integer satisfying $s \geq 2$. Thus, each of the N − 1 absorbing state rows $\bar{n}_A \in S_A' - \{\bar{m}_A\}$ of $\left|\bar{Q}_{\bar{m}_A}\right|$ can be written as a sum of two rows, one row containing $-ML\alpha/(1+\alpha)^{ML}$ at its principal diagonal location and zeros elsewhere and the second row being a multiple of $\alpha^2/(1+\alpha)^{ML}$. It follows from elementary determinant row expansion operations that $\left|\bar{P}_{\bar{m}_A} - \bar{I}\right| = \left|\bar{Q}_{\bar{m}_A}\right|$ can be written as

$$\left|\bar{P}_{\bar{m}_A} - \bar{I}\right| = \left|\bar{Q}_{\bar{m}_A}\right| = \frac{(-ML\alpha)^{N-1}\left|\bar{Q}_{\bar{m}_A}'\right|}{(1+\alpha)^{ML(N-1)}} + \frac{O(\alpha^s)}{(1+\alpha)^{MLN}} \qquad \text{Eq. 7.6}$$

where $\left|\bar{Q}_{\bar{m}_A}'\right|$ is the order N' − N + 1 principal minor of $\left|\bar{Q}_{\bar{m}_A}\right|$ generated by deleting the N − 1 row/column pairs which intersect on the $\bar{n}_A \in S_A' - \{\bar{m}_A\}$ principal diagonals and where the exponent s of α in the Eq. 7.6 order expression is an integer satisfying $s \geq N$.

The elements in all rows of $\left|\bar{Q}_{\bar{m}_A}'\right|$ except row $\bar{m}_A$ are composed of contributions from the elements in nonabsorbing state rows of $\bar{P}$ and the −1 principal diagonal term contributed by $\bar{I}$ in $\left|\bar{P}_{\bar{m}_A} - \bar{I}\right|$. Row $\bar{m}_A$ of $\left|\bar{Q}_{\bar{m}_A}'\right|$ contains −1 at its principal diagonal location and zeros elsewhere. Thus, if $\left|\bar{Q}_{\bar{m}_A}'\right|$ is written as $\left|\bar{Q}_{\bar{m}_A}'\right| = \left|\bar{P}_{\bar{m}_A}' - \bar{I}'\right|$, then from the

recipe employed in its construction, it follows that the square matrix $\bar{P}_{\bar{m}_A}$' thus defined has dimension $N' - N + 1$ and that its elements are given by

Eq. 7.7

$$\forall \bar{m}, \bar{n} \in S' - S_A' + \{\bar{m}_A\} : P_{\bar{m}_A}(\bar{m} \mid \bar{n})' = \begin{cases} 0 & \bar{n} = \bar{m}_A \\ P(\bar{m} \mid \bar{n}) & \bar{n} \neq \bar{m}_A \\ & \bar{m} \in S' - S_A' + \\ & \{\bar{m}_A\} + S(\bar{m}_A)' - S_A''. \\ P(\bar{m} \mid \bar{n}) + \frac{1}{L} P(\bar{n}_A \mid \bar{n}) & \bar{n} \neq \bar{m}_A \\ & \bar{m} \in S(\bar{n}_A)' \\ & \bar{n}_A \in S_A' - \{\bar{m}_A\} \end{cases}$$

Careful examination of Eq. 7.7 reveals that the transformation by which $\bar{P}_{\bar{m}_A}$' is generated from $\bar{P}_{\bar{m}_A}$ preserves all row sums. Thus, $\bar{P}_{\bar{m}_A}$' is very similar in form to $\bar{P}_{\bar{m}_A}$. It is derived from a (fictitious) row stochastic matrix by setting a specified row (row $\bar{m}_A$) to zero.

If the preceding steps are repeated for $|\bar{P}_{\bar{m}} - \bar{I}|$ where $\bar{m} \notin S_A'$, except that all N absorbing state columns $\bar{n}_A \in S_A'$ are coalesced rather than just the $N - 1$ columns $\bar{n}_A \in S_A' - \{\bar{m}_A\}$, a result very similar in form to Eq. 7.6 obtains. That is,

$$|\bar{P}_{\bar{m}} - \bar{I}| = |\bar{Q}_{\bar{m}}| = \frac{(-ML\alpha)^N |\bar{Q}_{\bar{m}}'|}{(1+\alpha)^{MLN}} + \frac{O(\alpha^s)}{(1+\alpha)^{MLN'}} \qquad \text{Eq. 7.8}$$

where $|\bar{Q}_{\bar{m}}'|$ is the order $N' - N$ principal minor of $|\bar{Q}_{\bar{m}}|$ generated by deleting the N absorbing state row/column pairs and s is an integer satisfying $s \geq N + 1$. The nonabsorbing state row $\bar{m}$ contains -1 at its principal diagonal location and zeros elsewhere.

Substitution of Eq. 7.6 and 7.8 into Proposition 6.7 yields a form more amenable to examination of the $\alpha \to 0^+$ limiting stationary distribution. The two cases $\bar{m} \in S_A'$ and $\bar{m} \in S' - S_A'$ must be distinguished. Then, after some straightforward algebra,

$$q_\alpha(\bar{m}) = \begin{cases} \dfrac{\left|\bar{Q}_{\bar{m}_A}'\right| + O(\alpha)/(1+\alpha)^{N'-N+1}}{\sum_{\bar{n}_A \in S_A'} \left|\bar{Q}_{\bar{n}_A}'\right| + O(\alpha)/(1+\alpha)^{N'-N+1}} & \bar{m} = \bar{m}_A \in S_A' \\ \dfrac{O(\alpha)/(1+\alpha)^{N'-N+1}}{\sum_{\bar{n}_A \in S_A'} \left|\bar{Q}_{\bar{n}_A}'\right| + O(\alpha)/(1+\alpha)^{N'-N+1}} & \bar{m} \in S' - S_A' \end{cases}.$$

An equivalent result expressed in terms of the auxiliary matrices $\bar{P}_{\bar{m}_A}'$ is

$$q_\alpha(\bar{m}) = \begin{cases} \dfrac{\left|\bar{P}_{\bar{m}_A}' - \bar{I}'\right| + O(\alpha)/(1+\alpha)^{N'-N+1}}{\sum_{\bar{n}_A \in S_A'} \left|\bar{P}_{\bar{n}_A}' - \bar{I}'\right| + O(\alpha)/(1+\alpha)^{N'-N+1}} & \bar{m} = \bar{m}_A \in S_A' \\ \dfrac{O(\alpha)/(1+\alpha)^{N'-N+1}}{\sum_{\bar{n}_A \in S_A'} \left|\bar{P}_{\bar{n}_A}' - \bar{I}'\right| + O(\alpha)/(1+\alpha)^{N'-N+1}} & \bar{m} \in S' - S_A' \end{cases}. \qquad \text{Eq. 7.9}$$

By retracing the preceding steps by which $\bar{P}_{\bar{m}_A}$ was transformed into $\bar{P}_{\bar{m}_A}'$, companion results to Eq. 7.7 and Eq. 7.9 can be developed for $\bar{P}(\bar{m}_A)'$. The companion to Eq. 7.7 differs only in the elements of row $\bar{n} = \bar{m}_A$. Thus, if $P(\bar{m} \mid \bar{n})'$ denotes the general element in $\bar{P}(\bar{m}_A)'$, then

Eq. 7.10

$$\forall \bar{m}, \bar{n} \in S' - S_A' + \{\bar{m}_A\} : P(\bar{m} \mid \bar{n})' = \begin{cases} P(\bar{m} \mid \bar{n}) & \bar{m} \in S' - S_A' + \{\bar{m}_A\} + S(\bar{m}_A)' - S_A'' \\ P(\bar{m} \mid \bar{n}) + \dfrac{1}{L} P(\bar{n}_A \mid \bar{n}) & \bar{m} \in S(\bar{n}_A)' \\ & \bar{n}_A \in S_A' - \{\bar{m}_A\} \end{cases}.$$

Further, examination of Eq. 7.10 reveals that the row sum constraint on $\bar{P}$ is preserved in the transformation by which $\bar{P}(\bar{m}_A)'$ is generated (i.e. $\bar{P}(\bar{m}_A)'$ is a stochastic matrix). Thus, $|\bar{P}(\bar{m}_A)' - \bar{I}'| = 0$. A consequence is the Proposition 6.8 counterpart of Eq. 7.9,

Eq. 7.11

$$q_\alpha(\bar{m}) = \begin{cases} \dfrac{|\bar{P}(\bar{m}_A)' - \bar{I}'| - \left|\bar{P}_{\bar{m}_A}' - \bar{I}'\right| + O(\alpha)/(1+\alpha)^{N'-N+1}}{\sum_{\bar{n}_A \in S_A'} \left\{|\bar{P}(\bar{n}_A)' - \bar{I}'| - \left|\bar{P}_{\bar{n}_A}' - \bar{I}'\right|\right\} + O(\alpha)/(1+\alpha)^{N'-N+1}} & \bar{m} = \bar{m}_A \in S_A' \\ \dfrac{O(\alpha)/(1+\alpha)^{N'-N+1}}{\sum_{\bar{n}_A \in S_A'} \left\{|\bar{P}(\bar{n}_A)' - \bar{I}'| - \left|\bar{P}_{\bar{n}_A}' - \bar{I}'\right|\right\} + O(\alpha)/(1+\alpha)^{N'-N+1}} & \bar{m} \in S' - S_A' \end{cases}.$$

7.5 The Stationary Distribution Limit

The zero mutation probability limits of Eq. 7.9 and Eq. 7.11 exist if the determinant sums in the denominators are nonzero. In fact they are nonzero, as demonstrated in the following. This argument is very similar in form to the development in Section 6.3 concerning positivity of the stationary distribution. The essential step is demonstration of the existence of a primitive stochastic matrix $\overline{Q}'$ which satisfies both

$$\overline{0} \leq \lim_{\alpha \to 0^+} \overline{P}_{\overline{m}_A}' \leq \overline{Q}' \text{ and } \overline{Q}' \neq \lim_{\alpha \to 0^+} \overline{P}_{\overline{m}_A}'.$$

If the two-operator algorithm is under consideration, then the elements of the $\alpha \to 0^+$ limit of $\overline{P}_{\overline{m}_A}'$ are obtained by substituting the one-operator results in Eq. 4.2-5 into Eq. 7.7. If the three-operator case is under consideration, then Eq. 4.22 and Eq. 4.24,25 are employed. In the following, the two-operator notation is employed.

Let $\overline{Q}'$ be generated from the $\alpha \to 0^+$ limit of $\overline{P}_{\overline{m}_A}'$ by replacing row $\overline{m}_A$ with the row whose elements are given by

$$\forall \overline{m} \in S' - S_A' + \{\overline{m}_A\} : Q'(\overline{m} \mid \overline{m}_A) = \frac{1}{N' - N + 1} > 0. \qquad \text{Eq. 7.12}$$

Thus, the row sum of row $\overline{m}_A$ in $\overline{Q}'$ is 1. Since all remaining rows of $\overline{Q}'$ are identical to those of the $\alpha \to 0^+$ limit of $\overline{P}_{\overline{m}_A}'$, and consequently have row sum 1 by Eq. 7.7, $\overline{Q}'$ is a stochastic matrix. Additionally, it satisfies both

$$\overline{0} \leq \lim_{\alpha \to 0^+} \overline{P}_{\overline{m}_A}' \leq \overline{Q}' \text{ and } \overline{Q}' \neq \lim_{\alpha \to 0^+} \overline{P}_{\overline{m}_A}'.$$

$\overline{Q}'$ can be regarded as the state transition matrix of a fictitious Markov chain defined on the state space $S' - S_A' + \{\overline{m}_A\}$. Since $\overline{Q}(\overline{m}_A \mid \overline{m}_A) > 0$, the fictitious Markov chain is both aperiodic (Definition A9, Theorem A2) and primitive (Definition B1, Theorem B1) provided that it is irreducible (Definitions A7 and A8, Theorem A1). Thus, primitivity is established by demonstrating that every state $\overline{m} \in S' - S_A' + \{\overline{m}_A\}$ is accessible in some finite number of transitions from every state $\overline{n} \in S' - S_A' + \{\overline{m}_A\}$. Since all states in $S' - S_A' + \{\overline{m}_A\}$ are accessible in one transition from $\overline{m}_A$ (Eq. 7.12), it is

sufficient to demonstrate that $\overline{m}_A$ is accessible in some finite number of transitions from every state $\overline{n} \in S' - S_A'$.

Let $i_A \in S$ be the bit-string represented in $\overline{m}_A$, let $\overline{n} \in S' - S_A'$ and let $i_1 \in S$ be selected such that $n(i_1) > 0$ and $H(i_1, i_A) \leq H(i, i_A)$ for all i represented in $\overline{n}$. Then, two cases must be examined. In case (1), $i_1 = i_A$ while $i_1 \neq i_A$ for case (2).

If $i_1 = i_A$, it follows from Eq. 7.7 and the construction of $\overline{Q}'$ that

$$Q'(\overline{m}_A \mid \overline{n}) = \lim_{\alpha \to 0^+} P_{\overline{m}_A}(\overline{m}_A \mid \overline{n}) = \lim_{\alpha \to 0^+} P_2(\overline{m}_A \mid \overline{n})$$

$$= \left[\lim_{\alpha \to 0^+} P_2(i_A \mid \overline{n}) \right]^M$$

$$= P_1(i_A \mid \overline{n})^M = P_1(i_1 \mid \overline{n})^M > 0$$

and consequently $\overline{m}_A$ is accessible from $\overline{n}$ in 1 transition. Otherwise

$$\exists i_2 \in S(i_1) \ni H(i_2, i_A) = H(i_1, i_A) - 1$$

and further if $\overline{n}_1 \in S_A'$ is the one-operator absorbing state defined by the condition $n_1(i_1) = M$ while $\overline{n}_{12} \in S(\overline{n}_1)'$ is the adjacent nonabsorbing state defined by $n_{12}(i_1) = M - 1$, $n_{12}(i_2) = 1$, then from Eq. 7.7 and the construction of $\overline{Q}'$

$$Q'(\overline{n}_{12} \mid \overline{n}) = \lim_{\alpha \to 0^+} P_2(\overline{n}_{12} \mid \overline{n}) + \frac{1}{L} \lim_{\alpha \to 0^+} P_2(\overline{n}_1 \mid \overline{n})$$

$$= P_1(\overline{n}_{12} \mid \overline{n}) + \frac{1}{L} P_1(\overline{n}_1 \mid \overline{n})$$

$$= \frac{1}{L} P_1(\overline{n}_1 \mid \overline{n})$$

$$= \frac{1}{L} [P_1(i_1 \mid \overline{n})]^M > 0.$$

Thus, $\overline{n}_{12}$ is accessible from $\overline{n}$ in one transition. If $i_2 = i_A$, then by the case (1) argument $\overline{m}_A$ is accessible in one additional transition. Otherwise, the case (2) argument is repeated for some

$$i_3 \in S(i_2) \ni H(i_3, i_A) = H(i_2, i_A) - 1 = H(i_1, i_A) - 2.$$

This procedure necessarily terminates with $H(i_1, i_A) + 1$ applications and the corresponding state space trajectory is executed with nonzero probability.

From the foregoing argument, it follows that state $\overline{m}_A$ is accessible in some finite number of transitions from every state $\overline{n} \in S' - S_A'$, and thus that $\overline{Q}'$ is primitive. Then, since both

$$\overline{0} \leq \lim_{\alpha \to 0^+} \overline{P}_{\overline{m}_A}' \leq \overline{Q}' \text{ and } \overline{Q}' \neq \lim_{\alpha \to 0^+} \overline{P}_{\overline{m}_A}',$$

it follows from Theorem B4(e) that every eigenvalue of the $\alpha \to 0^+$ limit of $\overline{P}_{\overline{m}_A}'$ satisfies $|\lambda_i| < 1$. Proposition 7.4 below, which is the $\alpha \to 0^+$ counterpart of Proposition 6.9, is a consequence.

Proposition 7.4: The value of the determinant $\lim_{\alpha \to 0^+} \left| \overline{P}_{\overline{m}_A}' - \overline{I}' \right|$ satisfies

$$\forall \overline{m}_A \in S_A' : \quad (1) \quad \lim_{\alpha \to 0^+} \left| \overline{P}_{\overline{m}_A}' - \overline{I}' \right| \neq 0$$

$$(2) \quad \text{the algebraic sign of } \lim_{\alpha \to 0^+} \left| \overline{P}_{\overline{m}_A}' - \overline{I}' \right| \text{ is } (-1)^{N'-N+1}.$$

The conditions asserted in Proposition 7.4 ensure that substitution into Eq. 7.9 and 7.11 yields a determinate form in the $\alpha \to 0^+$ limit. Propositions 7.5 and 7.6 below represent the limiting forms, and consequently are respectively the limiting counterparts of Propositions 6.7 and 6.8.

Proposition 7.5: The components of $\lim_{\alpha \to 0^+} \overline{q}_\alpha$ exist and can be expressed in the form

$$\lim_{\alpha \to 0^+} q_\alpha(\overline{m}) = \begin{cases} \dfrac{\lim_{\alpha \to 0^+} \left| \overline{P}_{\overline{m}_A}' - \overline{I}' \right|}{\sum_{\overline{n}_A \in S_A'} \lim_{\alpha \to 0^+} \left| \overline{P}_{\overline{n}_A}' - \overline{I}' \right|} & \overline{m} = \overline{m}_A \in S_A' \\ 0 & \overline{m} \in S' - S_A' \end{cases}.$$

Proposition 7.6: The components of $\lim_{\alpha \to 0^+} \bar{q}_\alpha$ exist and can be expressed alternatively as

$$\lim_{\alpha \to 0^+} q_\alpha(\bar{m}) = q_0(\bar{m}) = \begin{cases} \dfrac{\lim_{\alpha \to 0^+} \left\{ |\bar{P}(\bar{m}_A)' - \bar{I}'| - \left|\bar{P}_{\bar{m}_A}' - \bar{I}'\right| \right\}}{\sum_{\bar{n}_A \in S_A'} \lim_{\alpha \to 0^+} \left\{ |\bar{P}(\bar{n}_A)' - \bar{I}'| - \left|\bar{P}_{\bar{n}_A}' - \bar{I}'\right| \right\}} & \bar{m} = \bar{m}_A \in S_A' \\ 0 & \bar{m} \in S' - S_A' \end{cases}.$$

An immediate consequence of Propositions 7.4 and 7.5 is strict positivity of the zero mutation probability limiting stationary distribution components for all absorbing state rows. That is, $\forall \bar{m}_A \in S_A' : q_0(\bar{m}_A) > 0$. The argument is analogous to that at the conclusion of Section 6.3 concerning strict positivity of all stationary distribution components when $\alpha > 0$. This result is anticipated by the simulation results in Section 5.5. A consequence is that the required limiting behavior for direct application of the simulated annealing convergence theory to the genetic algorithm model does not follow. However, the results displayed in Section 5.5 and developments produced in Section 9.3 suggest that the limiting distribution can be made arbitrarily close to the desired limiting behavior.

Since the $\alpha \to 0^+$ limit of the stationary distribution exists, the definition of $\bar{q}_\alpha$ can be extended to include the point $\alpha = 0$. That is

$$\bar{q}_\alpha|_{\alpha = 0} = \bar{q}_0 = \lim_{\alpha \to 0^+} \bar{q}_\alpha$$

where the values of the required limits are provided by Proposition 7.5. Proposition 7.7 below follows from this extended definition of $\bar{q}_\alpha$ and Proposition 7.2.

Proposition 7.7: For all $\alpha \geq 0$, the components of $\bar{q}_\alpha$ are continuous rational functions of the independent variable α.

Proposition 7.3, which concerns the first derivative of $\bar{q}_\alpha$, can also be extended to include the limiting case. The extension requires easily obtainable counterparts of Eq. 7.1-3 developed for $\left|\bar{P}_{\bar{m}_A}' - \bar{I}'\right|$ and Eq. 7.9. The Eq. 7.1 counterpart is

$$(1+\alpha)^{ML(N'-N)}\left|\bar{P}_{\bar{m}_A}' - \bar{I}'\right| = \theta_{\bar{m}_A}(\alpha)', \qquad \text{Eq. 7.13}$$

and that for Eq. 7.2 is

$$q_\alpha(\bar{m}) = \begin{cases} \dfrac{\theta_{\bar{m}_A}(\alpha)' + O(\alpha)}{\Theta(\alpha)' + O(\alpha)} & \bar{m} = \bar{m}_A \in S_A' \\ \dfrac{O(\alpha)}{\Theta(\alpha)' + O(\alpha)} & \bar{m} \in S' - S_A' \end{cases} \qquad \text{Eq. 7.14}$$

where $\Theta(\alpha)'$ is the polynomial counterpart (summed over $\bar{n}_A \in S_A'$) of $\Theta(\alpha)$ in Eq. 7.2. Differentiating Eq. 7.14 with respect to α yields a rational function with denominator polynomial $[\Theta(\alpha)' + O(\alpha)]^2$, whose $\alpha \to 0^+$ limit is nonzero by Proposition 7.4, Eq. 7.13 and the definition of $\Theta(\alpha)'$. Proposition 7.8 below follows from Proposition 7.3 and these observations.

Proposition 7.8: The components of the first derivative of $\bar{q}_\alpha$ with respect to α possess limits as $\alpha \to 0^+$.

Thus, a zero mutation probability limit exists for the time-homogeneous two and three-operator algorithm variants. The limit is represented by Propositions 7.5 and 7.6. Further, Propositions 7.7 and 7.8 establish some useful ancillary results concerning the stationary distribution behavior at the point $\alpha = 0$. These latter results are employed in the following section in establishing strong ergodicity of the inhomogeneous genetic algorithm Markov chain. Propositions 7.5 and 7.6 are used in Section 9 to develop a methodology for representing the stationary distribution limit.

SECTION 8

A MONOTONIC MUTATION PROBABILITY ERGODICITY BOUND

8.1 Overview

The annealing schedule bounds for the simulated annealing algorithm, which are reviewed in Section 2.4.2, are derived by requiring that the nonstationary Markov chain which represents the algorithm be strongly ergodic (Definition A13) and then deducing a monotonic lower bound on the algorithm control parameter. The methodology consists of demonstrating that the time-homogeneous Markov chain corresponding to every positive algorithm control parameter value possesses a stationary distribution, that the sequence of stationary distributions corresponding to any sequence of positive control parameter values converges to a limiting distribution if the control parameter sequence converges to zero, and then employing Definitions A11-A13 and Theorems A5-A7 to deduce a sufficient condition (the annealing schedule lower bound) to guarantee that the nonstationary algorithm achieves the limiting distribution (i.e. strong ergodicity).

The model development in Section 4 demonstrates that for all mutation probability values in the range $0 < p_m < 1$, the Markov chain representing either the two or three-operator time-homogeneous simple genetic algorithm possesses a stationary distribution. Section 7 demonstrates that the stationary distribution approaches a limit as the mutation probability parameter approaches zero. This section proposes and then verifies a monotone decreasing lower bound on the mutation probability sequence of the nonstationary genetic algorithm Markov chain which is sufficient to ensure strong ergodicity.

8.2 A Weak Ergodicity Bound

The following paragraphs propose and then verify a mutation probability parameter bound sufficient to ensure that the Markov chain of the corresponding nonstationary simple genetic algorithm is weakly ergodic (Definition A11). The bound applies to both the two and three-operator algorithms, and it appears in Proposition 8.1 below.

Proposition 8.1: The mutation probability bound given by

$$p_m(k) \geq \frac{1}{2} k^{-\frac{1}{ML}}$$

is sufficient to ensure weak ergodicity of the corresponding nonstationary (two or three-operator) simple genetic algorithm Markov chain.

This result is established by using the lower bounds on the two and three-operator conditional probabilities in Eq. 4.21 and 4.31 with Definitions A11 and A12 and Theorems A5 and A6. Applying the lower bound in Eq. 4.21 and 4.31 to $\tau_1(\cdot)$ of Definition A12 and Theorem A5 yields

$$\begin{aligned}
\tau_1(\overline{P}) &= 1 - \min_{\bar{n}_1, \bar{n}_2} \sum_{\bar{m}} \min(P(\bar{m} \mid \bar{n}_1), P(\bar{m} \mid \bar{n}_2)) \\
&\leq 1 - \sum_{\bar{m}} \binom{M}{\bar{m}} \left(\frac{\alpha}{1+\alpha} \right)^{ML} \\
&= 1 - \left(\frac{\alpha}{1+\alpha} \right)^{ML} \sum_{\bar{m}} \binom{M}{\bar{m}} \\
&= 1 - \left(\frac{\alpha}{1+\alpha} \right)^{ML} 2^{ML} \\
&= 1 - \left(\frac{2\alpha}{1+\alpha} \right)^{ML}.
\end{aligned}$$

Thus,

$$(1 - \tau_1(\overline{P})) \geq \left(\frac{2\alpha}{1+\alpha} \right)^{ML}$$

and consequently from Theorem A6, the chain is weakly ergodic if the sequence of control parameter values $\{\alpha(k)\}$ satisfies

$$\sum_{k=1}^{\infty}\left(\frac{2\alpha(k)}{1+\alpha(k)}\right)^{ML}=\infty.$$

Comparing this result to the known divergent series $\sum k^{-1}$, it follows that the Markov chain is weakly ergodic if the sequence $\{\alpha(k)\}$ satisfies

$$\left(\frac{2\alpha(k)}{1+\alpha(k)}\right)^{ML}\geq k^{-1},$$

from which

$$\left(\frac{\alpha(k)}{1+\alpha(k)}\right)\geq\frac{1}{2}k^{-\frac{1}{ML}}.$$

Using Eq. 4.13 to translate this result into an equivalent expression in $p_m(k)$ establishes Proposition 8.1.

8.3 Strong Ergodicity

The mutation probability schedule bound advanced in Proposition 8.1 is also sufficient to achieve strong ergodicity if it satisfies the condition on the sequence of vector differences in Theorem A7. The required sequence of vectors can be selected as the sequence of stationary distributions of the time-homogeneous Markov chains associated with the parameter sequence $\{p_m(k)\}$ (or equivalently with the corresponding sequence $\{\alpha(k)\}$).

Section 4 establishes that a stationary distribution exists for the time-homogeneous two and three-operator algorithms corresponding to every value of α satisfying $\alpha > 0$. Thus, associated with the sequence of control parameter values $\{\alpha(k)\}$ is a sequence of vectors $\{\bar{q}_k\}$ where $\bar{q}_k = \bar{q}_\alpha$ evaluated at $\alpha = \alpha(k)$. Further, based upon results established in Section 6, Section 7 demonstrates that an $\alpha \rightarrow 0^+$ limiting stationary distribution exists (Propositions 7.5 and 7.6), that the stationary distribution vector varies continuously for all α satisfying $\alpha \geq 0$ (Proposition 7.7) and that its first derivative exists and is continuous

for all α satisfying $\alpha > 0$ (Proposition 7.3). In particular, $\bar{q}_\alpha$ is continuous on the closed interval $0 \le \alpha \le 1$ and its first derivative exists at every interior point of that interval. Therefore, if consideration is limited to monotone decreasing control parameter sequences, then by the mean value theorem the difference between the $\overline{m}$ components of any two consecutive vectors in the sequence can be written as

$$q_{k+1}(\overline{m}) - q_k(\overline{m}) = \left[\frac{dq_\alpha(\overline{m})}{d\alpha} \right]_{\alpha = \alpha^*(k)} (\alpha(k+1) - \alpha(k))$$

where the value $\alpha^*(k)$ satisfies $\alpha(k+1) < \alpha^*(k) < \alpha(k)$. Consequently,

$$| q_{k+1}(\overline{m}) - q_k(\overline{m})| = \left| \left[\frac{dq_\alpha(\overline{m})}{d\alpha} \right]_{\alpha = \alpha^*(k)} \right| \times | \alpha(k+1) - \alpha(k)|$$

and

$$\sum_{k=1}^{\infty} | q_{k+1}(\overline{m}) - q_k(\overline{m})| = \sum_{k=1}^{\infty} \left\{ \left| \left[\frac{dq_\alpha(\overline{m})}{d\alpha} \right]_{\alpha = \alpha^*(k)} \right| \times | \alpha(k+1) - \alpha(k)| \right\}. \qquad \text{Eq. 8.1}$$

From Propositions 7.3 and 7.8, it is possible to define a function $g_\alpha(\overline{m})$ which is continuous in α on the closed interval $0 \le \alpha \le 1$ as follows

$$g_\alpha(\overline{m}) = \begin{cases} \dfrac{dq_\alpha(\overline{m})}{d\alpha} & 0 < \alpha \le 1 \\ \lim\limits_{\alpha \to 0^+} \dfrac{dq_\alpha(\overline{m})}{d\alpha} & \alpha = 0 \end{cases} . \qquad \text{Eq. 8.2}$$

Then, from a fundamental theorem in the calculus of functions, it follows that $g_\alpha(\overline{m})$ (and consequently that $| g_\alpha(\overline{m})|$) is bounded on the closed interval $0 \le \alpha \le 1$. Thus, if

$$B = \sup_{\overline{m} \in S^*, \alpha \in [0,1]} | g_\alpha(\overline{m})| , \qquad \text{Eq. 8.3}$$

then it follows from Eq. 8.2 that at every interior point of the interval $0 \le \alpha \le 1$

$$\left| \frac{dq_\alpha(\overline{m})}{d\alpha} \right| \le B,$$

and application of this result to Eq. 8.1 yields

$$\sum_{k=1}^{\infty} |q_{k+1}(\bar{m}) - q_k(\bar{m})| = \sum_{k=1}^{\infty} \left\{ \left| \left[\frac{dq_\alpha(\bar{m})}{d\alpha} \right]_{\alpha=\alpha^*(k)} \right| \times |\alpha(k+1) - \alpha(k)| \right\}$$

$$\leq \sum_{k=1}^{\infty} B \times |\alpha(k+1) - \alpha(k)| \qquad \text{Eq. 8.4}$$

$$= B \sum_{k=1}^{\infty} |\alpha(k+1) - \alpha(k)| .$$

Since only monotonic control parameter sequences are under consideration, the sum in the last line of Eq. 8.4 can be written as the difference of the initial and final parameter values of the sequence. Thus,

$$\sum_{k=1}^{\infty} |q_{k+1}(\bar{m}) - q_k(\bar{m})| \leq B(\alpha(1) - \alpha(\infty))$$

$$= B(1 - 0) \qquad \text{Eq. 8.4}$$

$$= B < \infty.$$

The series of vector differences required for application to Theorem A7 can then be written as

$$\sum_{k=1}^{\infty} |\bar{q}_{k+1} - \bar{q}_k|_1 = \sum_{k+1}^{\infty} \sum_{\bar{m} \in S'} |q_{k+1}(\bar{m}) - q_k(\bar{m})|$$

$$\leq \sum_{\bar{m} \in S'} B \qquad \text{Eq. 8.5}$$

$$= N'B < \infty.$$

Applying the combined results of Proposition 8.1 and Eq. 8.5 to Theorem A7 produces the goal of this section.

Proposition 8.2: The mutation probability bound given by

$$p_m(k) \geq \frac{1}{2} k^{-\frac{1}{ML}}$$

is sufficient to ensure strong ergodicity of the corresponding Markov chain. Further, the Markov chain representing any nonstationary two or three-operator simple genetic

algorithm for which the mutation probability sequence both observes this bound and converges to zero achieves (asymptotically) the limiting probability distribution defined in Propositions 7.5 and 7.6.

8.4 Comparison With the Simulated Annealing Parameter Bound

It is instructive to compare the mutation probability sequence bound developed here with the annealing schedule bounds reviewed in Section 2.4.2.2, both of which are of the form K/log(k). Let $\rho(k)$ be defined as the ratio

$$\rho(k) = p_m(k)/T(k)$$

where $p_m(k)$ is selected as the bound developed herein and T(k) is selected as the bound provided by either Eq. 2.12 or Eq. 2.13. That is

$$\rho(k) = \frac{1}{2}k^{-\frac{1}{ML}}/[K/\log(k)] \qquad \text{Eq. 8.5}$$

$$= \frac{1}{2K}\log(k)/k^{\frac{1}{ML}}.$$

Thus, decreasing values of $\rho(k)$ imply that the genetic algorithm convergence rate is superior (asymptotically) to that of the simulated annealing algorithm.

Now, let $k = \exp(x)$, or equivalently $x = \log(k)$. Substituting into Eq. 8.5 yields

$$\rho(k) = \frac{1}{2K}x\exp\left(-\frac{x}{ML}\right).$$

Then, since for all positive constants γ, the limit of $x\exp(-\gamma x)$ as $x \to \infty$ is zero, it follows that

$$\lim_{k\to\infty}\rho(k) = \frac{1}{2K}\lim_{k\to\infty}\left[\log(k)/k^{\frac{1}{ML}}\right] = 0. \qquad \text{Eq. 8.6}$$

Thus, the nonstationary simple genetic algorithm provides an asymptotically superior convergence rate.

SECTION 9

REPRESENTATION OF THE STATIONARY DISTRIBUTION SOLUTION

9.1 Overview

Previous sections of this work establish some key results required for extrapolation of the simulated annealing convergence theory onto the nonstationary Markov chain model of the simple genetic algorithm. Specifically, existence of a unique stationary distribution for the time-homogeneous two and three-operator algorithms is established in Sections 4.3.2 and 4.3.3, and in Section 6, the existence argument is formulated into a Cramer's rule expression for the stationary distribution components. Sections 7 and 8 continue that development by establishing the existence of a limiting distribution as the mutation probability parameter approaches zero and a mutation probability sequence bound sufficient to achieve it. However, the empirical results in Section 5.5 suggest a complication, confirmed in Section 7, associated with the form of the limiting distribution. The limiting distribution behavior necessary for extending the simulated annealing global optimality result does not obtain because the limiting distribution is nonzero for all states with uniform population (one-operator absorbing states), including those for suboptimal solutions. The limiting distribution entropy results reported at the conclusion of Section 5.5 support the intuitive notion that increasing the population size parameter should bias the limiting distribution toward the desired behavior. However, to pursue that notion further requires closer examination of the stationary distribution equations and the requirements for their solution. This section begins that task. It is a very extensive development and it stops short of explicit solution. However, it provides some insight into the nature of the solution and additionally, it defines a promising approach to continuing the work started here.

The essential task of representing the stationary distribution solution consists of evaluating the determinants required to express the results of Propositions 6.7-8 and their limiting counterparts in Propositions 7.5-6. The development proceeds by examining the three distinct cases which arise from applying three different sets of constraints on the value of the mutation probability parameter. The special case $p_m = 1/2 \Leftrightarrow \alpha = 1$ is examined in Section 9.2. It leads to a very simple (trivial) result that is of no particular interest in its own right, but is fundamental to the mechanism employed in Section 9.3 in developing the more general case $0 < \alpha \leq 1$. The approach pursued in Section 9.3 involves expanding $-|\overline{P}_{\overline{m}} - \overline{I}| = |\overline{P} - \overline{I}| - |\overline{P}_{\overline{m}} - \overline{I}|$ as a multivariate Taylor's series in the $N' \times N$ array of conditional probabilities $[P(i \mid \overline{n})]$ for $i \in S$ and $\overline{n} \in S'$ (defined by Eq. 4.14 for the two-operator algorithm and by Eq. 4.22 for its three-operator counterpart) about the point corresponding to $\alpha = 1$. The product of that effort is an expression for the coefficient of the general term of the series as a determinant with combinatorial elements. The case $p_m \rightarrow 0^+ \Leftrightarrow \alpha \rightarrow 0^+$ is examined in Section 9.4. The methodology developed in Section 9.3 extends with very little modification to represent the $\alpha \rightarrow 0^+$ limiting behavior of $|\overline{P}_{\overline{m}} - \overline{I}|$. Section 9.5 concludes by pointing out some significant identities which exist among the Taylor's series coefficients and the connection of those identities to the algebra of symmetric and alternating polynomials. Its purpose is to provide a foundation for extending the stationary distribution representation work begun here.

9.2 The Limiting Case $\alpha = 1$

As pointed out in Section 6, the determinants required for expressing the value of the stationary distribution components by Propositions 6.7-8 are the characteristic polynomials of the $\overline{P}_{\overline{m}}$ matrices evaluated at $\lambda = 1$. The coefficients of the characteristic polynomial of any square matrix $\overline{X}$ with finite dimensions can be expressed in terms of the principal minors of $|\overline{X}|$ (i.e. minors generated from $|\overline{X}|$ by deleting combinations of rows and columns with the same indices). For example, the characteristic polynomial of $\overline{P}$ can be expressed as

$$\begin{aligned}\phi(\lambda) &= |\bar{P} - \lambda\bar{I}| \\ &= A_{N'} - A_{N'-1}\lambda + A_{N'-2}\lambda^2 - A_{N'-3}\lambda^3 \\ &+ \cdots + (-1)^{N'-1}A_1\lambda^{N'-1} + (-1)^{N'}A_0\lambda^{N'} \\ &= \sum_u (-1)^{N'-u}A_u\lambda^{N'-u}\end{aligned} \qquad \text{Eq. 9.1}$$

where N' = card(S') is the dimension of $\bar{P}$ and A_u is the sum of its order u principal minors. This result is fundamental to the theory of square matrices and follows from application of elementary determinant expansion operations to $|\bar{P} - \lambda\bar{I}|$ [Aitk54, MoSt64, Muir60]. Exactly

$$\binom{N'}{u} = \frac{N'!}{u!(N'-u)!}$$

order u principal minors are summed to produce A_u. The values of some of the A_u's are

$$A_u = \begin{cases} |\bar{P}| & u = N' \\ \text{trace}(\bar{P}) & u = 1 \\ 1 & u = 0 \end{cases} \qquad \text{Eq. 9.2}$$

where the A_0 result follows from the convention that the single order zero principal minor of $|\bar{X}|$ has value 1.

In a fashion exactly analogous to Eq. 9.1, the characteristic polynomial of $\bar{P}_{\bar{m}}$ can be written as

$$\begin{aligned}\phi_{\bar{m}}(\lambda) &= |\bar{P}_{\bar{m}} - \lambda\bar{I}| \\ &= A^{\bar{m}}_{N'} - A^{\bar{m}}_{N'-1}\lambda + A^{\bar{m}}_{N'-2}\lambda^2 - A^{\bar{m}}_{N'-3}\lambda^3 \\ &+ \cdots + (-1)^{N'-1}A^{\bar{m}}_1\lambda^{N'-1} + (-1)^{N'}A^{\bar{m}}_0\lambda^{N'} \\ &= \sum_u (-1)^{N'-u}A^{\bar{m}}_u\lambda^{N'-u}\end{aligned} \qquad \text{Eq. 9.3}$$

where $A^{\bar{m}}_u$ is the sum of all order u principal minors of $|\bar{P}_{\bar{m}}|$. Thus, the value of each determinant required for expressing $\bar{q}$ via Propositions 6.7-8 can be written respectively in the form

$$|\bar{P}_{\bar{m}} - \bar{I}| = \phi_{\bar{m}}(1)$$

$$= A_{N'}^{\bar{m}} - A_{N'-1}^{\bar{m}} + A_{N'-2}^{\bar{m}} - A_{N'-3}^{\bar{m}}$$

$$+ \cdots + (-1)^{N'-1} A_1^{\bar{m}} + (-1)^{N'} A_0^{\bar{m}}$$

$$= \sum_u (-1)^{N'-u} A_u^{\bar{m}}$$

and

$$|\bar{P} - \bar{I}| - |\bar{P}_{\bar{m}} - \bar{I}| = \phi(1) - \phi_{\bar{m}}(1)$$

$$= (A_{N'} - A_{N'}^{\bar{m}}) - (A_{N'-1} - A_{N'-1}^{\bar{m}}) + (A_{N'-2} - A_{N'-2}^{\bar{m}}) - (A_{N'-3} - A_{N'-3}^{\bar{m}}) \qquad \text{Eq. 9.4}$$

$$+ \cdots + (-1)^{N'-1} (A_1 - A_1^{\bar{m}}) + (-1)^{N'} (A_0 - A_0^{\bar{m}})$$

$$= \sum_u (-1)^{N'-u} (A_u - A_u^{\bar{m}}).$$

Further, the $A_u^{\bar{m}}$'s can be expressed in terms of the principal minors of $|\bar{P}|$ because those principal minors of $|\bar{P}_{\bar{m}}|$ which include row $\bar{m}$ (the $\bar{0}^T$ row) have value zero while those which exclude row $\bar{m}$ are identical to the corresponding principal minors of $|\bar{P}|$. In particular, the $A_u^{\bar{m}}$'s corresponding to Eq. 9.2 are

$$A_u^{\bar{m}} = \begin{cases} |\bar{P}_{\bar{m}}| = 0 & u = N' \\ \text{trace}(\bar{P}_{\bar{m}}) = \text{trace}(\bar{P}) - P(\bar{m} \mid \bar{m}) & u = 1 \\ 1 & u = 0 \end{cases} . \qquad \text{Eq. 9.5}$$

From Eq. 4.11 and Eq. 4.24, it follows that

$$\forall i \in S, \forall \bar{n} \in S' : P(i \mid \bar{n})|_{\alpha=1} = \left(\frac{1}{(1+\alpha)^L} \times \sum_{j \in S} P(j \mid \bar{n}) \times \alpha^{H(i,j)} \right)_{\alpha=1}$$

$$= \frac{1}{2^L} \qquad \text{Eq. 9.6}$$

$$= \frac{1}{N}.$$

The conditional probabilities are uniform at $\alpha = 1$. Then, by substituting from Eq. 9.6 into Eq. 4.15 or 4.25, it follows that

$$\forall \bar{m}, \bar{n} \in S' : P(\bar{m} \mid \bar{n})|_{\alpha=1} = \binom{M}{\bar{m}} \prod_{i \in s} \left(\frac{1}{2^L}\right)^{m(i)} \qquad \text{Eq. 9.7}$$

$$= \binom{M}{\bar{m}} \frac{1}{2^{ML}}.$$

Thus, $\bar{P}|_{\alpha=1} = [P(\bar{m} \mid \bar{n})]_{\alpha=1}$ is a rank one matrix, and therefore all minors of $|\bar{P}|_{\alpha=1}$ of order $u \geq 2$ are identically zero. Eq. 9.4 then reduces to

$$\{|\bar{P} - \bar{I}| - |\bar{P}_{\bar{m}} - \bar{I}|\}_{\alpha=1} = \{\phi(1) - \phi_{\bar{m}}(1)\}_{\alpha=1}$$

$$= (-1)^{N'-1}\{A_1 - A_1^{\bar{m}}\}_{\alpha=1} + (-1)^{N'}\{A_0 - A_0^{\bar{m}}\}_{\alpha=1},$$

and substituting from Eq. 9.2, 9.5 and 9.7 into this result produces

$$\{|\bar{P} - \bar{I}| - |\bar{P}_{\bar{m}} - \bar{I}|\}_{\alpha=1} = (-1)^{N'-1}\{\text{trace}(\bar{P}) - [\text{trace}(\bar{P}) - P(\bar{m} \mid \bar{m})]\}_{\alpha=1}$$

$$= (-1)^{N'-1} P(\bar{m} \mid \bar{m})|_{\alpha=1} \qquad \text{Eq. 9.8}$$

$$= (-1)^{N'-1}\binom{M}{\bar{m}}\frac{1}{2^{ML}}.$$

Employing Eq. 9.8 with Proposition 6.8 yields an explicit result for the $\alpha = 1$ limiting value of $q(\bar{m})$, i.e.

$$q(\bar{m})|_{\alpha=1} = \left.\frac{|\bar{P} - \bar{I}| - |\bar{P}_{\bar{m}} - \bar{I}|}{\sum_{\bar{n} \in S'}(|\bar{P} - \bar{I}| - |\bar{P}_{\bar{n}} - \bar{I}|)}\right|_{\alpha=1}$$

$$= \frac{(-1)^{N'-1}\binom{M}{\bar{m}}/2^{ML}}{\sum_{\bar{n} \in S'}(-1)^{N'-1}\binom{M}{\bar{n}}/2^{ML}} = \frac{\binom{M}{\bar{m}}}{\sum_{\bar{n} \in S'}\binom{M}{\bar{n}}}$$

$$= \binom{M}{\bar{m}}\frac{1}{2^{ML}}.$$

It is independent of the objective function of the underlying optimization problem because at $p_m = 1/2 \Leftrightarrow \alpha = 1$ mutation completely nullifies the reproduction operator.

Although this trivial case is not of any particular interest on its own, it serves as the basis for developing the general case $0 < \alpha \leq 1$ in Section 9.3. The essential idea is that $\bar{P}$

is rank 1 at $\alpha = 1$, which makes the low order derivatives of $\bar{P}$ with respect to the conditional probabilities $P(i \mid \bar{n})$ have comparatively low rank, and this suggests expanding $-|\bar{P}_{\bar{m}} - \bar{I}| = |\bar{P} - \bar{I}| - |\bar{P}_{\bar{m}} - \bar{I}|$ in a multivariate Taylor's series about the point corresponding to $\alpha = 1$. The result reflected in Eq. 9.8 is the constant term of the series.

9.3 The General Case $0 < \alpha \leq 1$

The state transition matrix of the two and three-operator algorithms is completely determined by the fixed algorithm parameter M and the $N' \times N$ array of conditional probabilities $[P(i \mid \bar{n})]$ for $i \in S$ and $\bar{n} \in S'$. Each element of row $\bar{n}$ in $\bar{P}$ consists of a multinomial coefficient and a distinct order M product composed of integral powers of the $P(i \mid \bar{n})$'s corresponding to row $\bar{n}$ (Eq. 4.15 or 4.25). Thus, the order k principal minor of $|\bar{P}|$ generated by inclusion of rows $K = \{\bar{n}_1, \bar{n}_2, \cdots, \bar{n}_k\} \subseteq S'$ can be written as an order $k \times M$ polynomial (composed of order $k \times M$ monomial terms) in the $k \times N$ array of variables $[P(i \mid \bar{n})]$ for $i \in S, \bar{n} \in K$. The corresponding order k principal minor of $|\bar{P}_{\bar{m}}|$ has identical value provided $\bar{m} \notin K$ and is zero if $\bar{m} \in K$. These facts along with the succinct representation of the $P(i \mid \bar{n})$'s as rational functions of the objective function and algorithm parameters (Eq. 4.2, 4.11, 4.24) and the degeneration of $\bar{P}$ to rank 1 at $\alpha = 1$ (Eq. 9.7) suggest an attempt to expand $\phi_{\bar{m}}(1) = |\bar{P}_{\bar{m}} - \bar{I}|$ as a multivariate Taylor's series in the $P(i \mid \bar{n})$'s about the point $\alpha = 1$ where, according to Eq. 9.6, $\forall i \in S, \forall \bar{n} \in S' : P(i \mid \bar{n}) = 1/2^L$. (Actually, expanding the alternative form $\phi(1) - \phi_{\bar{m}}(1) = |\bar{P} - \bar{I}| - |\bar{P}_{\bar{m}} - \bar{I}|$ in a new array of $N' \times N$ variables which uniquely determines $[P(i \mid \bar{n})]$ proves more productive). The constant term in the series is provided by Eq. 9.8 and the highest order terms in the series are the order $(N' - 1) \times M$ monomials contributed by the single nonzero order $N' - 1$ principal minor of $|\bar{P}_{\bar{m}}|$.

Let $\bar{r} = [r(i, \bar{n})]$ be an $N' \times N$ nonnegative integer array having rows of the form $\bar{r}_{\bar{n}} = (r(0, \bar{n}), r(1, \bar{n}), \cdots, r((2^L - 1), \bar{n}))$. The nonnegative integer $r(i, \bar{n})$ represents the exponent of the factor $(P(i \mid \bar{n}) - 1/2^L)$ appearing in a monomial term of the Taylor's series.

Also, let $\|\bar{r}_{\bar{n}}\| = \sum_{i \in S} r(i,\bar{n})$ and $\|\bar{r}\| = \sum_{\bar{n} \in S'} \|\bar{r}_{\bar{n}}\| = \sum_{\bar{n} \in S'} \sum_{i \in S} r(i,\bar{n})$. Then, the Taylor's series expansion of $-|\bar{P}_{\bar{m}} - \bar{I}| = |\bar{P} - \bar{I}| - |\bar{P}_{\bar{m}} - \bar{I}|$ can be expressed as

$$-|\bar{P}_{\bar{m}} - \bar{I}| = |\bar{P} - \bar{I}| - |\bar{P}_{\bar{m}} - \bar{I}| \qquad \text{Eq. 9.9}$$

$$= \sum_{\bar{r}} C(\bar{m},\bar{r}) \times \prod_{\bar{n} \in S'} \prod_{i \in S} (P(i \mid \bar{n}) - 1/2^L)^{r(i,\bar{n})}$$

where

$$C(\bar{m},\bar{r}) = \left. \frac{\partial^{\|\bar{r}\|}(-|\bar{P}_{\bar{m}} - \bar{I}|)}{\prod_{\bar{n} \in S'} \prod_{i \in S} r(i,\bar{n})! \partial^{r(i,\bar{n})} P(i,\bar{n})} \right|_{\alpha=1} = \frac{(-|\bar{P}_{\bar{m}} - \bar{I}|)^{(\bar{r})}|_{\alpha=1}}{\bar{r}!} \qquad \text{Eq. 9.10}$$

$$= \left. \frac{\partial^{\|\bar{r}\|}(|\bar{P} - \bar{I}| - |\bar{P}_{\bar{m}} - \bar{I}|)}{\prod_{\bar{n} \in S'} \prod_{i \in S} r(i,\bar{n})! \partial^{r(i,\bar{n})} P(i \mid \bar{n})} \right|_{\alpha=1} = \frac{(|\bar{P} - \bar{I}| - |\bar{P}_{\bar{m}} - \bar{I}|)^{(\bar{r})}|_{\alpha=1}}{\bar{r}!}$$

is the coefficient of the order $\|\bar{r}\|$ monomial term uniquely identified by the nonnegative integer array $\bar{r}$. In these expressions, the symbol $\bar{r}!$ denotes the operation

$$\bar{r}! = \prod_{\bar{n} \in S'} \prod_{i \in S} [r(i,\bar{n})!].$$

Expressing the value of $C(\bar{m},\bar{r})$ thus reduces to evaluating the indicated mixed partial derivative of $-|\bar{P}_{\bar{m}} - \bar{I}| = |\bar{P} - \bar{I}| - |\bar{P}_{\bar{m}} - \bar{I}|$ at $\alpha = I$ divided by $\bar{r}!$.

The coefficient of the order $\|\bar{r}\| = \|\bar{0}\| = 0$ term is

$$C(\bar{m},\bar{0}) = (|\bar{P} - \bar{I}| - |\bar{P}_{\bar{m}} - \bar{I}|)^{(\bar{0})}\Big|_{\alpha=1} = \{|\bar{P} - \bar{I}| - |\bar{P}_{\bar{m}} - \bar{I}|\}_{\alpha=1}$$

and its value is the constant term of the series, provided by Eq. 9.8. The coefficient of the first order monomial term which results from setting

$$\bar{r}^{(1)} = [r(i,\bar{n})^{(1)}] \quad \text{where} \quad r(i,\bar{n})^{(1)} = \begin{cases} 1 & i = i_1, \bar{n} = \bar{n}_1 \\ 0 & \text{otherwise} \end{cases}$$

is given by

$$C(\bar{m},\bar{r}^{(1)}) = \left. \frac{\partial(|\bar{P} - \bar{I}| - |\bar{P}_{\bar{m}} - \bar{I}|)}{\partial P(i_1 \mid \bar{n}_1)} \right|_{\alpha=1}$$

$$= (|\bar{P} - \bar{I}| - |\bar{P}_{\bar{m}} - \bar{I}|)^{(\bar{r}^{(1)})}\Big|_{\alpha=1}.$$

The associated monomial term in the Taylor's series expansion of $-|\bar{P}_{\bar{m}} - \bar{I}| = |\bar{P} - \bar{I}| - |\bar{P}_{\bar{m}} - \bar{I}|$ is given by

$$C(\bar{m}, \bar{r}^{(1)}) \times (P(i_1 \mid \bar{n}_1) - 1/2^L).$$

In subsequent paragraphs, $\|\bar{r}_{\bar{m}}\| = 0$ and $\|\bar{r}_{\bar{n}}\| \leq M$ for $\bar{n} \neq \bar{m}$, which together imply $\|\bar{r}\| \leq (N' - 1) \times M$, are shown to be suitable upper bounds on the order of differentiation with respect to the $P(i \mid \bar{n})$'s when computing $C(\bar{m}, \bar{r})$. Thus, the Taylor's series terminates at finite order, as indeed it must since $-|\bar{P}_{\bar{m}} - \bar{I}|$ is a polynomial function of the $P(i \mid \bar{n})$'s, and as noted earlier, the highest order monomial terms are order $(N' - 1) \times M$.

These upper bounds on the order of differentiation (i.e. upper bounds on $\|\bar{r}_{\bar{n}}\|$), along with the lower bound of 0 on every component of $\bar{r}$ imposed by the requirement that $\bar{r}$ be a nonnegative integer array, can be represented to advantage in terms of a set related to S'. Let S', which is completely determined by the parameters L and M, be momentarily represented by S'(M) (that is, let its dependence on M be explicitly indicated) and let the set S'' be defined as the set union of all S'(k) for $0 \leq k \leq M$. That is

$$S'' = S'(0) \cup S'(1) \cup \cdots \cup S'(M-1) \cup S'(M)$$

$$= \bigcup_{k=0}^{M} S'(k).$$

The above constraints on the rows of $\bar{r}$ are then equivalent to requiring that every row of $\bar{r}$ be drawn from S'', with the additional requirement that row $\bar{m}$ be the specific element $\bar{r}_{\bar{m}} = \bar{0} \in S''$. Since the S'(k) for distinct k are disjoint, it follows that $N'' = \text{card}(S'')$ is the sum of the $N'(k) = \text{card}[S'(k)]$, and consequently from Eq. 4.1 and an elementary recursion on the binomial coefficient that

$$N'' = \binom{M + 2^L}{M} = \binom{M+N}{M}.$$

This result is precisely that supplied by Eq. 5.1, accompanying the state space enumeration empirical results tabulated in Section 5.2.

Since $\overline{P}_{\overline{m}}$ is independent of the $P(i \mid \overline{m})$'s associated with row $\overline{m}$ (row $\overline{m}$ of $\overline{P}_{\overline{m}}$ is set to $\overline{0}^T$), it follows that no monomial term containing any of the factors $(P(i \mid \overline{m}) - 1/2^L)$ appears in the expansion of $-|\overline{P}_{\overline{m}} - \overline{I}|$. Further, since $|\overline{P} - \overline{I}| - |\overline{P}_{\overline{m}} - \overline{I}| = -|\overline{P}_{\overline{m}} - \overline{I}|$, no such monomial terms appear with nonzero coefficient in the expansion of $|\overline{P} - \overline{I}| - |\overline{P}_{\overline{m}} - \overline{I}|$ either. (Due to the constraint $\forall \overline{n} \in S' : \sum_{i \in S} P(i \mid \overline{n}) = 1$, the aggregate of such terms appearing in $|\overline{P} - \overline{I}|$ is identically zero). This observation establishes the $\|\overline{r}_{\overline{m}}\| = 0$ bound and permits the following revision of the Eq. 9.10 definition of $C(\overline{m}, \overline{r})$

$$C(\overline{m}, \overline{r}) = \begin{cases} \dfrac{(|\overline{P} - \overline{I}| - |\overline{P}_{\overline{m}} - \overline{I}|)^{(\overline{r})}|_{\alpha=1}}{\overline{r}!} & \|\overline{r}_{\overline{m}}\| = 0 \\ 0 & \text{otherwise} \end{cases} . \qquad \text{Eq. 9.11}$$

The derivative of an order n determinant with respect to a variable x can be written as the sum of the n determinants generated by differentiating each row (or column) in turn with respect to x [Aitk54, MoSt64, Muir60]. For example, if

$$\overline{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix},$$

then

$$\frac{d|\overline{A}|}{dx} = \begin{vmatrix} \dfrac{da_{11}}{dx} & \dfrac{da_{12}}{dx} \\ a_{21} & a_{22} \end{vmatrix} + \begin{vmatrix} a_{11} & a_{12} \\ \dfrac{da_{21}}{dx} & \dfrac{da_{22}}{dx} \end{vmatrix}.$$

If the elements of any row in the given determinant are independent of x, then differentiation of that row introduces an all zero row and the value of the corresponding determinant is zero. In particular, if only one row of the given determinant depends upon x, then only one nonzero determinant appears in the row-derivative expansion.

Higher order and mixed partial derivatives of an order n determinant can be expressed similarly, e.g.

$$\frac{\partial^3|\bar{A}|}{\partial x_1^2\partial x_2}=\begin{vmatrix}\frac{\partial^3 a_{11}}{\partial x_1^2\partial x_2} & \frac{\partial^3 a_{12}}{\partial x_1^2\partial x_2}\\ a_{21} & a_{22}\end{vmatrix}+\begin{vmatrix}\frac{\partial^2 a_{11}}{\partial x_1^2} & \frac{\partial^2 a_{12}}{\partial x_1^2}\\ \frac{\partial a_{21}}{\partial x_2} & \frac{\partial a_{22}}{\partial x_2}\end{vmatrix}$$

$$+2\begin{vmatrix}\frac{\partial^2 a_{11}}{\partial x_1\partial x_2} & \frac{\partial^2 a_{12}}{\partial x_1\partial x_2}\\ \frac{\partial a_{21}}{\partial x_1} & \frac{\partial a_{22}}{\partial x_1}\end{vmatrix}+2\begin{vmatrix}\frac{\partial a_{11}}{\partial x_1} & \frac{\partial a_{12}}{\partial x_1}\\ \frac{\partial^2 a_{21}}{\partial x_1\partial x_2} & \frac{\partial^2 a_{22}}{\partial x_1\partial x_2}\end{vmatrix}+$$

$$\begin{vmatrix}\frac{\partial a_{11}}{\partial x_2} & \frac{\partial a_{12}}{\partial x_2}\\ \frac{\partial^2 a_{21}}{\partial x_1^2} & \frac{\partial^2 a_{22}}{\partial x_1^2}\end{vmatrix}+\begin{vmatrix}a_{11} & a_{12}\\ \frac{\partial^3 a_{21}}{\partial x_1^2\partial x_2} & \frac{\partial^3 a_{22}}{\partial x_1^2\partial x_2}\end{vmatrix},$$

and again, differentiation of any row with respect to a variable upon which it does not depend introduces an all zero row. Thus, if in the preceding result the first row of $\bar{A}$ is independent of x_2 and the second of x_1, then only one of the determinants in the expansion survives

$$\frac{\partial^3|\bar{A}|}{\partial x_1^2\partial x_2}=\begin{vmatrix}\frac{\partial^2 a_{11}}{\partial x_1^2} & \frac{\partial^2 a_{12}}{\partial x_1^2}\\ \frac{\partial a_{21}}{\partial x_2} & \frac{\partial a_{22}}{\partial x_2}\end{vmatrix}.$$

Since each $P(i\,|\,\bar{n})$ appears in only one row of $-|\bar{P}_{\bar{m}}-\bar{I}|=|\bar{P}-\bar{I}|-|\bar{P}_{\bar{m}}-\bar{I}|$, it follows from application of the preceding determinant differentiation rules that the mixed partial derivative $|\bar{P}-\bar{I}|^{(\bar{r})}$ can be written as the single determinant (indicated hereafter by $|(\bar{P}-\bar{I})^{(\bar{r})}|=|\bar{P}^{(\bar{r})}-\bar{I}^{(\bar{r})}|$) generated by differentiating the rows of the matrix $(\bar{P}-\bar{I})$ in accordance with $\bar{r}$ and then computing the determinant of the matrix derivative. That is, due to the single-row dependence of $(\bar{P}-\bar{I})$ on each $P(i\,|\,\bar{n})$, the two operations involved (differentiating $(\bar{P}-\bar{I})$ and evaluating its determinant) commute. The same conclusion applies to any mixed partial derivative of $|\bar{P}_{\bar{m}}-\bar{I}|$ with respect to the $P(i\,|\,\bar{n})$'s, and hence

$$(|\bar{P}-\bar{I}|-|\bar{P}_{\bar{m}}-\bar{I}|)^{(\bar{r})}=|\bar{P}-\bar{I}|^{(\bar{r})}-|\bar{P}_{\bar{m}}-\bar{I}|^{(\bar{r})}$$

$$=|(\bar{P}-\bar{I})^{(\bar{r})}|-\left|(\bar{P}_{\bar{m}}-\bar{I})^{(\bar{r})}\right| \qquad \text{Eq. 9.12}$$

$$=|\bar{P}^{(\bar{r})}-\bar{I}^{(\bar{r})}|-\left|\bar{P}_{\bar{m}}^{(\bar{r})}-\bar{I}^{(\bar{r})}\right|.$$

Eq. 9.4 can be generalized to express the value of $(|\bar{P}-\bar{I}|-|\bar{P}_{\bar{m}}-\bar{I}|)^{(\bar{r})}$ as indicated in the following. Let $\bar{r}$ have $k \le N'-1$ rows which specify nonzero order differentiation, let $K=\{\bar{n}_1,\bar{n}_2,\cdots,\bar{n}_k\}\subset S'$ be the set of differentiated-row indices and further let $\bar{m}\notin K$. Also, for $N' \ge u \ge k$ let $A_u(\bar{r})$ be the sum of all order u principal minors of $|\bar{P}^{(\bar{r})}|$ formed by including the k differentiated rows indicated by $\bar{n}\in K$ and $u-k$ of the $N'-k$ undifferentiated rows in $|\bar{P}^{(\bar{r})}|$. Exactly

$$\binom{N'-k}{u-k}=\frac{(N'-k)!}{(u-k)!(N'-u)!}$$

order u principal minors are summed to produce $A_u(\bar{r})$. Finally, let $A_u^{(\bar{m})}(\bar{r})$ be defined similarly for $|\bar{P}_{\bar{m}}^{(\bar{r})}|$. Then, applying the same elementary determinant expansion rules that lead to Eq. 9.1 and Eq. 9.3 to $|\bar{P}^{(\bar{r})}-\lambda\bar{I}^{(\bar{r})}|$ and $|\bar{P}_{\bar{m}}^{(\bar{r})}-\lambda\bar{I}^{(\bar{r})}|$ yields

$$|\bar{P}^{(\bar{r})}-\lambda\bar{I}^{(\bar{r})}|=\sum_{u=k}^{N'}(-1)^{N'-u}A_u(\bar{r})\lambda^{N'-u}$$

and

$$\left|\bar{P}_{\bar{m}}^{(\bar{r})}-\lambda\bar{I}^{(\bar{r})}\right|=\sum_{u=k}^{N'}(-1)^{N'-u}A_u^{\bar{m}}(\bar{r})\lambda^{N'-u},$$

and substituting these results into Eq. 9.12 with $\lambda=1$ yields the differentiated analog of Eq. 9.4

$$(|\bar{P}-\bar{I}|-|\bar{P}_{\bar{m}}-\bar{I}|)^{(\bar{r})}=|\bar{P}^{(\bar{r})}-\bar{I}^{(\bar{r})}|-\left|\bar{P}_{\bar{m}}^{(\bar{r})}-\bar{I}^{(\bar{r})}\right| \qquad \text{Eq. 9.13}$$

$$=\sum_{u=k}^{N'}(-I)^{N'-u}(A_u(\bar{r})-A_u^{\bar{m}}(\bar{r})).$$

If $|\bar{P}^{(\bar{r})}|_K$ is the order k principal minor of $|\bar{P}^{(\bar{r})}|$ uniquely defined by the set of row/column indices $K=\{\bar{n}_1,\bar{n}_2,\cdots,\bar{n}_k\}\subset S'$ where $\bar{m}\notin K$ and if $|\bar{P}^{(\bar{r})}|_{K_{\bar{n}}}$ is the order $k+1$ principal minor generated by including the undifferentiated row $\bar{n}\in S'-K$ with K (i.e.

$K_{\bar{n}} = K \cup \bar{n} = \{\bar{n}_1, \bar{n}_2, \cdots, \bar{n}_k, \bar{n}\} \subseteq S')$, then

$$A_u(\bar{r}) = \begin{cases} |\bar{P}^{(\bar{r})}|_{S'} = |\bar{P}^{(\bar{r})}| & u = N' \\ \sum_{\bar{n} \in S'-K} |\bar{P}^{(\bar{r})}|_{K_{\bar{n}}} & u = k+1 \\ |\bar{P}^{(\bar{r})}|_K & u = k \end{cases} . \qquad \text{Eq. 9.14}$$

Also, since every principal minor of $|\bar{P}^{(\bar{r})}_{\bar{m}}|$ is either identical to the corresponding principal minor of $|\bar{P}^{(\bar{r})}|$ or 0 depending upon whether or not it includes row $\bar{m}$, and since $\bar{m} \notin K$ by hypothesis, it follows that the $A^{\bar{m}}_u(\bar{r})$'s corresponding to Eq. 9.14 are

$$A^{\bar{m}}_u(\bar{r}) = \begin{cases} \left|\bar{P}^{(\bar{r})}_{\bar{m}}\right|_{S'} = \left|\bar{P}^{(\bar{r})}_{\bar{m}}\right| = 0 & u = N' \\ \sum_{\bar{n} \in S'-K-\{\bar{m}\}} |\bar{P}^{(\bar{r})}|_{K_{\bar{n}}} = A_{k+1}(\bar{r}) - |\bar{P}^{(\bar{r})}|_{K_{\bar{m}}} & u = k+1 \\ |\bar{P}^{(\bar{r})}|_K & u = k \end{cases} . \qquad \text{Eq. 9.15}$$

All of the $N' - k$ undifferentiated rows in $\bar{P}^{(\bar{r})}|_{\alpha=1}$ are identical (Eq. 9.7) and consequently all minors of $\bar{P}^{(\bar{r})}|_{\alpha=1}$ of order $u \geq k+2$ have value 0. Thus, in a fashion exactly analogous to the derivation of Eq. 9.8 from Eq. 9.4, Eq. 9.13 yields

$$\left\{ (|\bar{P} - \bar{I}| - |\bar{P}_{\bar{m}} - \bar{I}|)^{(\bar{r})} \right\}_{\alpha=1} = (-1)^{N'-k-1} \{A_{k+1}(\bar{r}) - A^{\bar{m}}_{k+1}(\bar{r})\}_{\alpha=1} + (-1)^{N'-k} \{A_k(\bar{r}) - A^{\bar{m}}_k(\bar{r})\}_{\alpha=1},$$

and substituting from Eq. 9.14 and 9.15 into this result produces

$$\left\{ (|\bar{P} - \bar{I}| - |\bar{P}_{\bar{m}} - \bar{I}|)^{(\bar{r})} \right\}_{\alpha=1} = (-1)^{N'-k-1} \left\{ A_{k+1}(\bar{r}) - \left[A_{k+1}(\bar{r}) - |\bar{P}^{(\bar{r})}|_{K_{\bar{m}}} \right] \right\}_{\alpha=1} \qquad \text{Eq. 9.16}$$

$$= (-1)^{N'-k-1} \left\{ |\bar{P}^{(\bar{r})}|_{K_{\bar{m}}} \right\}_{\alpha=1}$$

from which, by substitution into Eq. 9.11, it follows that

$$C(\bar{m}, \bar{r}) = \begin{cases} \dfrac{(-1)^{N'-k-1} \left\{ |\bar{P}^{(\bar{r})}|_{K_{\bar{m}}} \right\}_{\alpha=1}}{\bar{r}!} & \|\bar{r}_{\bar{m}}\| = 0 \\ 0 & \text{otherwise} \end{cases} . \qquad \text{Eq. 9.17}$$

Evaluating $C(\bar{m}, \bar{r})$ thus requires evaluating the quotient of the order $k+1$ principal minor $\left\{ |\bar{P}^{(\bar{r})}|_{K_{\bar{m}}} \right\}_{\alpha=1}$ and $\bar{r}!$.

The order $k+1$ principal minor $|\bar{P}^{(\bar{r})}|_{K_{\bar{m}}}$ is completely determined by the $(k+1) \times (k+1)$ sub-array of $\bar{P}^{(\bar{r})}$ given by $[P^{(\bar{r})}(\bar{w} \mid \bar{v})]$ for $\bar{w}, \bar{v} \in K_{\bar{m}}$ where $P^{(\bar{r})}(\bar{w} \mid \bar{v})$

denotes the indicated mixed partial derivative of $P(\overline{w} \mid \overline{v})$. Further, in computing the determinant of this sub-array, the order in which the row/column indices are drawn from $K_{\overline{m}}$ in the sub-array construction is immaterial because any transposition of the order introduces exactly one row transposition and one column transposition into the sub-array, so both the magnitude and the algebraic sign of its determinant are preserved. Thus, the most general form of $|\overline{P}^{(r)}|_{K_{\overline{m}}}$ can be expressed as

$$|\overline{P}^{(r)}|_{K_{\overline{m}}} = \begin{vmatrix} P^{(r)}(\overline{m} \mid \overline{m}) & P^{(r)}(\overline{n}_1 \mid \overline{m}) & P^{(r)}(\overline{n}_2 \mid \overline{m}) & \cdots & P^{(r)}(\overline{n}_k \mid \overline{m}) \\ P^{(r)}(\overline{m} \mid \overline{n}_1) & P^{(r)}(\overline{n}_1 \mid \overline{n}_1) & P^{(r)}(\overline{n}_2 \mid \overline{n}_1) & \cdots & P^{(r)}(\overline{n}_k \mid \overline{n}_1) \\ P^{(r)}(\overline{m} \mid \overline{n}_2) & P^{(r)}(\overline{n}_1 \mid \overline{n}_2) & P^{(r)}(\overline{n}_2 \mid \overline{n}_2) & \cdots & P^{(r)}(\overline{n}_k \mid \overline{n}_2) \\ . & . & . & \cdots & . \\ . & . & . & \cdots & . \\ . & . & . & \cdots & . \\ P^{(r)}(\overline{m} \mid \overline{n}_k) & P^{(r)}(\overline{n}_1 \mid \overline{n}_k) & P^{(r)}(\overline{n}_2 \mid \overline{n}_k) & \cdots & P^{(r)}(\overline{n}_k \mid \overline{n}_k) \end{vmatrix}. \qquad \text{Eq. 9.18}$$

From Eq. 4.15 and 4.25, it follows that each nonzero element in row $\overline{v} \in K_{\overline{m}}$ of $\overline{P}^{(r)}$ is composed of a combinatorial coefficient and an order $M - \|\overline{r}_{\overline{v}}\|$ product of the $P(i \mid \overline{v})$'s. The general form of the element in column $\overline{w} \in K_{\overline{m}}$ of row $\overline{v}$ is given by

$$P^{(r)}(\overline{w} \mid \overline{v}) = \begin{cases} \binom{M}{\overline{w}} \dfrac{\prod_{i \in S} w(i)!}{\prod_{i \in S} (w(i) - r(i,\overline{v}))!} \prod_{i \in S} P(i \mid \overline{v})^{w(i) - r(i,\overline{v})} & \forall i \in S : w(i) \geq r(i,\overline{v}) \\ 0 & \text{otherwise} \end{cases},$$

which can be rewritten as

Eq. 9.19

$$P^{(r)}(\overline{w} \mid \overline{v}) = \begin{cases} \binom{M}{\overline{w}} \left[\dfrac{\prod_{i \in S} w(i)! \times \left(\prod_{i \in S} [r(i,\overline{v})!] P(i \mid \overline{v})^{w(i) - r(i,\overline{v})} \right)}{\prod_{i \in S} r(i,\overline{v})! (w(i) - r(i,\overline{v}))!} \right] & \forall i \in S : w(i) \geq r(i,\overline{v}) \\ 0 & \text{otherwise} \end{cases}.$$

Further, by noting that the factor

$$\left.\begin{aligned}\binom{M}{\bar{w}}\left[\frac{\prod_{i\in S} w(i)!}{\prod_{i\in S} r(i,\bar{v})!(w(i)-r(i,\bar{v}))!}\right] \\ = \frac{M!}{\prod_{i\in S} w(i)!}\times\left[\frac{\prod_{i\in S} w(i)!}{\prod_{i\in S} r(i,\bar{v})!(w(i)-r(i,\bar{v}))!}\right] \\ = \frac{M!}{\prod_{i\in S} r(i,\bar{v})!(w(i)-r(i,\bar{v}))!}\end{aligned}\right\} \quad \forall i \in S: w(i) \geq r(i,\bar{v})$$

is a multinomial coefficient and designating it (via straightforward generalization of the convention introduced in Eq. 4.4) by

$$\binom{M}{\bar{w},\bar{r}_{\bar{v}}} = \begin{cases} \dfrac{M!}{\prod_{i\in S}(w(i)-r(i,\bar{v}))!r(i,\bar{v})!} & \forall i \in S: w(i) \geq r(i,\bar{v}) \\ 0 & \text{otherwise} \end{cases}, \qquad \text{Eq. 9.20}$$

Eq. 9.19 simplifies to

$$P^{(r)}(\bar{w} \mid \bar{v}) = \binom{M}{\bar{w},\bar{r}_{\bar{v}}} \prod_{i\in S} [r(i,\bar{v})! P(i \mid \bar{v})^{w(i)-r(i,\bar{v})}]. \qquad \text{Eq. 9.21}$$

If row $\bar{v}$ is undifferentiated (i.e. $\|\bar{r}_{\bar{v}}\| = 0$), then Eq. 9.21 becomes

$$P^{(r)}(\bar{w} \mid \bar{v}) = \binom{M}{\bar{w}} \prod_{i\in S} P(i \mid \bar{v})^{w(i)} = P(\bar{w} \mid \bar{v}).$$

It is noted in passing that if $M \leq N = 2^L$, then it follows from Eq. 9.20 that either

$$\binom{M}{\bar{w},\bar{r}_{\bar{v}}} = 0$$

or Eq. 9.22

$$\exists \bar{w}' \in S' \ni \binom{M}{\bar{w},\bar{r}_{\bar{v}}} = \binom{M}{\bar{w}'}.$$

In the latter case, it is also true that

$$\binom{M}{\bar{w}'} \geq \binom{M}{\bar{w}}.$$

The enabling condition imposes no practical limitation because any algorithm with $M \geq N$ could be effectively supplanted by exhaustive search over S.

Since $|\bar{P}^{(\bar{r})}|_{K_{\bar{m}}}$ includes every row of $\bar{P}^{(\bar{r})}$ which is differentiated to nonzero order (e.g. $\forall \bar{n} \in S' \ni \|\bar{r}_{\bar{n}}\| > 0$), it follows from Eq. 9.18, Eq. 9.20 and Eq. 9.21 that any row $\bar{n}$ for which $\|\bar{r}_{\bar{n}}\| > M$ introduces an all zero row into $|\bar{P}^{(\bar{r})}|_{K_{\bar{m}}}$, making both $|\bar{P}^{(\bar{r})}|_{K_{\bar{m}}} = 0$ and $C(\bar{m},\bar{r}) = 0$. Therefore, $\|\bar{r}_{\bar{n}}\| \leq M$ represents a suitable upper bound on $\|\bar{r}_{\bar{n}}\|$ for $\bar{n} \neq \bar{m}$. This bound, along with the previously established condition $\|\bar{r}_{\bar{m}}\| = 0$, implies $\|\bar{r}\| \leq (N' - 1) \times M$ and permits the following revision of the Eq. 9.17 definition of $C(\bar{m},\bar{r})$

$$C(\bar{m},\bar{r}) = \begin{cases} \dfrac{(-1)^{N'-k-1}\left\{|\bar{P}^{(\bar{r})}|_{K_{\bar{m}}}\right\}_{\alpha=1}}{\bar{r}!} & \|\bar{r}_{\bar{m}}\| = 0, \|\bar{r}_{\bar{n}}\| \leq M \text{ for } \bar{n} \neq \bar{m} \\ 0 & \text{otherwise} \end{cases} .$$

Further, the conditions in this result can be expressed in terms of S'', yielding

$$C(\bar{m},\bar{r}) = \begin{cases} \dfrac{(-1)^{N'-k-1}\left\{|\bar{P}^{(\bar{r})}|_{K_{\bar{m}}}\right\}_{\alpha=1}}{\bar{r}!} & \bar{r}_{\bar{m}} = \bar{0}, \bar{r}_{\bar{n}} \in S'' - \{\bar{0}\} \text{ for } \bar{n} \neq \bar{m} \\ 0 & \text{otherwise} \end{cases} . \quad \text{Eq. 9.23}$$

At $\alpha = 1$, using Eq. 9.6 in Eq. 9.21 yields -

$$P^{(\bar{r})}(\bar{w} \mid \bar{v})|_{\alpha=1} = \binom{M}{\bar{w}, \bar{r}_{\bar{v}}} \prod_{i \in S} [r(i,\bar{v})!(1/2^L)^{w(i)-r(i,\bar{v})}] \quad \text{Eq. 9.24}$$

$$= \binom{M}{\bar{w}, \bar{r}_{\bar{v}}} (1/2^L)^{M-|\bar{r}_{\bar{v}}|} \prod_{i \in S} [r(i,\bar{v})!].$$

Thus, every element in row $\bar{v} \in K_{\bar{m}}$ of $|\bar{P}^{(\bar{r})}|_{K_{\bar{m}}}$ includes the constant factor

$$(1/2^L)^{M-|\bar{r}_{\bar{v}}|} \prod_{i \in S} [r(i,\bar{v})!].$$

Substituting Eq. 9.24 into Eq. 9.18 and collecting these common row factors outside the determinant yields

$$\left\{ \left| \overline{P}^{(\bar{r})} \right|_{K_{\bar{m}}} \right\}_{\alpha=1} = (1/2^L)^{M(k+1)-|\bar{r}|} \times \left[\prod_{\bar{v} \in K_{\bar{m}}} \prod_{i \in S} r(i,\bar{v})! \right]$$

$$\times \begin{vmatrix} \binom{M}{\bar{m}, \bar{r}_{\bar{m}}} & \binom{M}{\bar{n}_1, \bar{r}_{\bar{m}}} & \binom{M}{\bar{n}_2, \bar{r}_{\bar{m}}} & \cdots & \binom{M}{\bar{n}_k, \bar{r}_{\bar{m}}} \\ \binom{M}{\bar{m}, \bar{r}_{\bar{n}_1}} & \binom{M}{\bar{n}_1, \bar{r}_{\bar{n}_1}} & \binom{M}{\bar{n}_2, \bar{r}_{\bar{n}_1}} & \cdots & \binom{M}{\bar{n}_k, \bar{r}_{\bar{n}_1}} \\ \binom{M}{\bar{m}, \bar{r}_{\bar{n}_2}} & \binom{M}{\bar{n}_1, \bar{r}_{\bar{n}_2}} & \binom{M}{\bar{n}_2, \bar{r}_{\bar{n}_2}} & \cdots & \binom{M}{\bar{n}_k, \bar{r}_{\bar{n}_2}} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \binom{M}{\bar{m}, \bar{r}_{\bar{n}_k}} & \binom{M}{\bar{n}_1, \bar{r}_{\bar{n}_k}} & \binom{M}{\bar{n}_2, \bar{r}_{\bar{n}_k}} & \cdots & \binom{M}{\bar{n}_k, \bar{r}_{\bar{n}_k}} \end{vmatrix}.$$

Also, since $\bar{r}_{\bar{m}} = \bar{0} \Rightarrow \| \bar{r}_{\bar{m}} \| = 0 \Rightarrow \binom{M}{\bar{w}, \bar{r}_{\bar{m}}} = \binom{M}{\bar{w}, \bar{0}} = \binom{M}{\bar{w}}$ and since $\prod_{\bar{v} \in K_{\bar{m}}} \prod_{i \in S} r(i,\bar{v})! = \bar{r}!$,

$$\left\{ \left| \overline{P}^{(\bar{r})} \right|_{K_{\bar{m}}} \right\}_{\alpha=1} = (1/2^L)^{M(k+1)-|\bar{r}|} \times \bar{r}! \times \begin{vmatrix} \binom{M}{\bar{m}} & \binom{M}{\bar{n}_1} & \binom{M}{\bar{n}_2} & \cdots & \binom{M}{\bar{n}_k} \\ \binom{M}{\bar{m}, \bar{r}_{\bar{n}_1}} & \binom{M}{\bar{n}_1, \bar{r}_{\bar{n}_1}} & \binom{M}{\bar{n}_2, \bar{r}_{\bar{n}_1}} & \cdots & \binom{M}{\bar{n}_k, \bar{r}_{\bar{n}_1}} \\ \binom{M}{\bar{m}, \bar{r}_{\bar{n}_2}} & \binom{M}{\bar{n}_1, \bar{r}_{\bar{n}_2}} & \binom{M}{\bar{n}_2, \bar{r}_{\bar{n}_2}} & \cdots & \binom{M}{\bar{n}_k, \bar{r}_{\bar{n}_2}} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \binom{M}{\bar{m}, \bar{r}_{\bar{n}_k}} & \binom{M}{\bar{n}_1, \bar{r}_{\bar{n}_k}} & \binom{M}{\bar{n}_2, \bar{r}_{\bar{n}_k}} & \cdots & \binom{M}{\bar{n}_k, \bar{r}_{\bar{n}_k}} \end{vmatrix},$$

and substitution into Eq. 9.23 yields

$$C(\bar{m},\bar{r}) = (-1)^{N'-k-1} \times (1/2^L)^{M(k+1)-|\bar{r}|} \qquad \text{Eq. 9.25}$$

$$\times \begin{vmatrix} \binom{M}{\bar{m}} & \binom{M}{\bar{n}_1} & \binom{M}{\bar{n}_2} & \cdots & \binom{M}{\bar{n}_k} \\ \binom{M}{\bar{m},\bar{r}_{\bar{n}_1}} & \binom{M}{\bar{n}_1,\bar{r}_{\bar{n}_1}} & \binom{M}{\bar{n}_2,\bar{r}_{\bar{n}_1}} & \cdots & \binom{M}{\bar{n}_k,\bar{r}_{\bar{n}_1}} \\ \binom{M}{\bar{m},\bar{r}_{\bar{n}_2}} & \binom{M}{\bar{n}_1,\bar{r}_{\bar{n}_2}} & \binom{M}{\bar{n}_2,\bar{r}_{\bar{n}_2}} & \cdots & \binom{M}{\bar{n}_k,\bar{r}_{\bar{n}_2}} \\ . & . & . & \cdots & . \\ . & . & . & \cdots & . \\ . & . & . & \cdots & . \\ \binom{M}{\bar{m},\bar{r}_{\bar{n}_k}} & \binom{M}{\bar{n}_1,\bar{r}_{\bar{n}_k}} & \binom{M}{\bar{n}_2,\bar{r}_{\bar{n}_k}} & \cdots & \binom{M}{\bar{n}_k,\bar{r}_{\bar{n}_k}} \end{vmatrix}.$$

Note that the condition $\bar{r}_{\bar{m}} = \bar{0}$ is implicitly asserted in this result by the form of the first row of the combinatorial determinant, and that the condition $\bar{r}_{\bar{n}} \in S'' - \{\bar{0}\}$ for $\bar{n} \neq \bar{m}$ is enforced by the definition in Eq. 9.20.

When Eq. 9.25 is employed with Eq. 9.9, an additional simplification becomes available. The simplification obtains by incorporating the factor

$$(1/2^L)^{-|\bar{r}|} = 2^{L|\bar{r}|}$$

present in the Eq. 9.24 definition of $C(\bar{m},\bar{r})$ with the product factor in Eq. 9.9. That is

$$-|\bar{P}_{\bar{m}} - \bar{I}| = |\bar{P} - \bar{I}| - |\bar{P}_{\bar{m}} - \bar{I}| = \sum_{\bar{r}} C(\bar{m},\bar{r}) \times \prod_{\bar{n} \in S'} \prod_{i \in S} (P(i \mid \bar{n}) - 1/2^L)^{r(i,\bar{n})}$$

$$= \sum_{\bar{r}} C(\bar{m},\bar{r}) \times \left(\frac{1}{2^L}\right)^{|\bar{r}|} \times (2^L)^{|\bar{r}|} \prod_{\bar{n} \in S'} \prod_{i \in S} (P(i \mid \bar{n}) - 1/2^L)^{r(i,\bar{n})} \qquad \text{Eq. 9.26}$$

$$= \sum_{\bar{r}} C'(\bar{m},\bar{r}) \times \prod_{\bar{n} \in S'} \prod_{i \in S} (2^L P(i \mid \bar{n}) - 1)^{r(i,\bar{n})}$$

where

$$C'(\bar{m},\bar{r}) = C(\bar{m},\bar{r}) \times (1/2^L)^{|\bar{r}|}$$

is the coefficient of the indicated monomial in the new variables. Substitution of Eq. 9.25 into this expression for $C'(\bar{m},\bar{r})$ yields

$$C'(\overline{m},\bar{r}) = (-1)^{N'-k-1} \times \left(\frac{1}{2^{ML}}\right)^{(k+1)} \qquad \text{Eq. 9.27}$$

$$\times \begin{vmatrix} \binom{M}{\overline{m}} & \binom{M}{\overline{n}_1} & \binom{M}{\overline{n}_2} & \cdots & \binom{M}{\overline{n}_k} \\ \binom{M}{\overline{m},\bar{r}_{\overline{n}_1}} & \binom{M}{\overline{n}_1,\bar{r}_{\overline{n}_1}} & \binom{M}{\overline{n}_2,\bar{r}_{\overline{n}_1}} & \cdots & \binom{M}{\overline{n}_k,\bar{r}_{\overline{n}_1}} \\ \binom{M}{\overline{m},\bar{r}_{\overline{n}_2}} & \binom{M}{\overline{n}_1,\bar{r}_{\overline{n}_2}} & \binom{M}{\overline{n}_2,\bar{r}_{\overline{n}_2}} & \cdots & \binom{M}{\overline{n}_k,\bar{r}_{\overline{n}_2}} \\ . & . & . & \cdots & . \\ . & . & . & \cdots & . \\ . & . & . & \cdots & . \\ \binom{M}{\overline{m},\bar{r}_{\overline{n}_k}} & \binom{M}{\overline{n}_1,\bar{r}_{\overline{n}_k}} & \binom{M}{\overline{n}_2,\bar{r}_{\overline{n}_k}} & \cdots & \binom{M}{\overline{n}_k,\bar{r}_{\overline{n}_k}} \end{vmatrix}.$$

The special case $\bar{r} = \bar{0} \Leftrightarrow k = 0 \Leftrightarrow K_{\overline{m}} = \{\overline{m}\}$ of this result is $C'(\overline{m},\bar{0})$, the coefficient of the single order 0 (constant) monomial term previously represented by Eq. 9.8.

Eq. 9.26 and 9.27 represent an exact expression of the value of $-|\overline{P}_{\overline{m}} - \bar{I}| = |\overline{P} - \bar{I}| - |\overline{P}_{\overline{m}} - \bar{I}|$, and with Propositions 6.7 and 6.8 constitute an exact representation of the components of the stationary distributions of the two and three-operator algorithms. The utility of the representation depends upon the succinct representation of the components of the array $[P(i \mid \overline{n})]$ as rational functions of the algorithm parameters and objective function (Eq. 4.2, 4.11, 4.22 and 4.24) and the ability to extract useful relationships between the $C'(\overline{m},\bar{r})$'s from the general form represented by Eq. 9.27. Section 9.5 below points out some key coefficient identities related to the latter task, but first the results of this section are extended to include the limiting case represented by Propositions 7.5 and 7.6.

9.4 The Limiting Case $\alpha \to 0^+$

The matrices introduced in Section 7.4 for use in evaluating the stationary distribution zero mutation probability limit, $\overline{P}(\overline{m}_A)'$ and $\overline{P}_{\overline{m}_A}'$, are very similar in form to $\overline{P}$ and $\overline{P}_{\overline{m}_A}$. Specifically, $\overline{P}(\overline{m}_A)'$ is a row stochastic matrix and $\overline{P}_{\overline{m}_A}'$ is generated from it by

setting row $\bar{m}_A$ to $\bar{0}^T$. Further, by virtue of their construction (Eq. 7.7 and 7.10), the single row dependence of the matrix elements on the conditional probability array $[P(i \mid \bar{n})]$ employed in developing the results in Section 9.3 for $\bar{P}$ and $\bar{P}_{\bar{m}}$ applies to $\bar{P}(\bar{m}_A)'$ and $\bar{P}_{\bar{m}_A}'$ as well. Thus, Eq. 9.26-27 should extend with very little modification to the determinants $-\left|\bar{P}_{\bar{m}_A}' - \bar{I}'\right| = |\bar{P}(\bar{m}_A)' - \bar{I}'| - \left|\bar{P}_{\bar{m}_A}' - \bar{I}'\right|$, whose zero mutation probability limits are required by Propositions 7.5-6. The following paragraphs highlight the required modifications and employ the result to examine two simple examples.

In the $\alpha \to 0^+$ counterpart of Eq. 9.26-27, $\bar{m}$ is limited to membership in the set of one-operator absorbing states (i.e. $\bar{m} = \bar{m}_A \in S_A'$), a consequence of which is that

$$\binom{M}{\bar{m}_A} = 1.$$

Also, all rows of the determinant other than $\bar{m}_A$ correspond to nonabsorbing states (i.e. $\bar{n} \in K \subset S' - S_A'$). Thus, the determinant order is $N' - N + 1$ and the differentiation index array is order $(N' - N + 1) \times N$ with rows corresponding to row indices $\bar{n} \in S' - S_A' + \{\bar{m}_A\}$. The rows of $\bar{r}$ are limited to $\bar{r}_{\bar{m}_A} = \bar{0}$ and $\bar{r}_{\bar{n}} \in S''$ for $\bar{n} \in S' - S_A$. Further, if $\bar{r}$ indicates nonzero order differentiation of any rows which are adjacent to one-operator absorbing states, then the associated columns of the combinatorial determinant must reflect the coefficient contribution from the adjacent absorbing state. Thus, if $C_0'(\bar{m}_A, \bar{r})$ denotes the limiting counterpart of $C'(\bar{m}, \bar{r})$ in Eq. 9.27 and if $K = \{\bar{n}_1, \bar{n}_2, \cdots, \bar{n}_k\} \subset S' - S_A'$ where (in the state adjacency notation introduced in Section 7.3) $\bar{n}_1 \in S(\bar{n}_{A_1})' \neq S(\bar{m}_A)'$ and where $\bar{n}_2, \cdots, \bar{n}_k$ all satisfy $\bar{n}_j \notin S_A''$, then the coefficient of the order k monomial term uniquely identified by $\bar{r}$ is given by

$$C_0'(\bar{m}_A,\bar{r}) = (-1)^{N'-N-k} \times \left(\frac{1}{2^{ML}}\right)^{(k+1)} \qquad \text{Eq. 9.28}$$

$$\times \begin{vmatrix} 1 & \binom{M}{\bar{n}_1} + \frac{1}{L} & \binom{M}{\bar{n}_2} & \cdots & \binom{M}{\bar{n}_k} \\ \binom{M}{\bar{m}_A, \bar{r}_{\bar{n}_1}} & \binom{M}{\bar{n}_1, \bar{r}_{\bar{n}_1}} + \frac{1}{L}\binom{M}{\bar{n}_{A_1}, \bar{r}_{\bar{n}_1}} & \binom{M}{\bar{n}_2, \bar{r}_{\bar{n}_1}} & \cdots & \binom{M}{\bar{n}_k, \bar{r}_{\bar{n}_1}} \\ \binom{M}{\bar{m}_A, \bar{r}_{\bar{n}_2}} & \binom{M}{\bar{n}_1, \bar{r}_{\bar{n}_2}} + \frac{1}{L}\binom{M}{\bar{n}_{A_1}, \bar{r}_{\bar{n}_2}} & \binom{M}{\bar{n}_2, \bar{r}_{\bar{n}_2}} & \cdots & \binom{M}{\bar{n}_k, \bar{r}_{\bar{n}_2}} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \binom{M}{\bar{m}_A, \bar{r}_{\bar{n}_k}} & \binom{M}{\bar{n}_1, \bar{r}_{\bar{n}_k}} + \frac{1}{L}\binom{M}{\bar{n}_{A_1}, \bar{r}_{\bar{n}_k}} & \binom{M}{\bar{n}_2, \bar{r}_{\bar{n}_k}} & \cdots & \binom{M}{\bar{n}_k, \bar{r}_{\bar{n}_k}} \end{vmatrix}.$$

It is noted that Eq. 9.28 is only an example, not a definition. It must be adjusted based upon $\bar{r}$ to reflect the number and location of the nonzero adjacent state contributions.

The values of the determinants $-\left|\bar{P}_{\bar{m}_A}' - \bar{I}'\right| = |\bar{P}(\bar{m}_A)' - \bar{I}'| - \left|\bar{P}_{\bar{m}_A}' - \bar{I}'\right|$ are given by employing Eq. 9.28 in Eq. 9.26 (with $\bar{r}$ restricted as noted above). Further, the $\alpha \to 0^+$ limits of the $-\left|\bar{P}_{\bar{m}_A}' - \bar{I}'\right| = |\bar{P}(\bar{m}_A)' - \bar{I}'| - \left|\bar{P}_{\bar{m}_A}' - \bar{I}'\right|$ are provided by using the $\alpha \to 0^+$ limits of the factors $(2^L P(i \mid \bar{n}) - 1)$ in Eq. 9.26. Those limits are provided by using either $P_1(i \mid \bar{n})$ or $P_2'(i \mid \bar{n})$ depending upon whether the two or three-operator case is under consideration.

It is instructive to apply these results to a simple example. The following paragraphs do so for the one-bit problem with population size 2. These parameters $(L = 1, M = 2)$ imply that $S = \{0, 1\}$, $N = 2$, $S' = \{(20), (11), (02)\}$, $N' = 3$, $S_A' = \{(20), (02)\}$ and $S' - S_A' = \{(11)\}$. Thus $\bar{r}$ is limited to

$$\bar{r} \in \left\{ \begin{pmatrix} 00 \\ 00 \\ 00 \end{pmatrix}, \begin{pmatrix} 00 \\ 10 \\ 00 \end{pmatrix}, \begin{pmatrix} 00 \\ 01 \\ 00 \end{pmatrix}, \begin{pmatrix} 00 \\ 11 \\ 00 \end{pmatrix}, \begin{pmatrix} 00 \\ 20 \\ 00 \end{pmatrix}, \begin{pmatrix} 00 \\ 02 \\ 00 \end{pmatrix} \right\},$$

and the combinatorial determinant required for evaluation of the nonzero order $C_0'(\bar{m}_A, \bar{r})$'s for $\bar{m}_A = (20)$ by Eq. 9.28 has the general form

$$\begin{vmatrix} \binom{2}{(20)} & \binom{2}{(11)}+\binom{2}{(02)} \\ \binom{2}{(20),\bar{r}_{11}} & \binom{2}{(11),\bar{r}_{11}}+\binom{2}{(02),\bar{r}_{11}} \end{vmatrix} = \begin{vmatrix} 1 & 2+1 \\ \binom{2}{(20),\bar{r}_{11}} & \binom{2}{(11),\bar{r}_{11}}+\binom{2}{(02),\bar{r}_{11}} \end{vmatrix}.$$

Evaluation of the zero order coefficient proceeds as follows

$$C_0'\left((20),\begin{pmatrix}00\\00\\00\end{pmatrix}\right) = (-1)^{(3-2-0)} \times \left(\frac{1}{2^2}\right)^1 \times \binom{2}{(20)}$$

$$= (-1) \times \frac{1}{4} \times 1$$

$$= -\frac{1}{4}.$$

The coefficient corresponding to $\bar{r}_{11} = (10)$ is given by

$$C_0'\left((20),\begin{pmatrix}00\\10\\00\end{pmatrix}\right) = (-1)^{(3-2-1)} \times \left(\frac{1}{2^2}\right)^{1+1} \times \begin{vmatrix} 1 & 2+1 \\ \binom{2}{(20),(10)} & \binom{2}{(11),(10)}+\binom{2}{(02),(10)} \end{vmatrix}$$

$$= \frac{1}{16} \times \begin{vmatrix} 1 & 3 \\ 2 & 2 \end{vmatrix}$$

$$= \frac{1}{16}(-4) = -\frac{1}{4},$$

and similarly for $\bar{r}_{11} = (01)$,

$$C_0'\left((20),\begin{pmatrix}00\\01\\00\end{pmatrix}\right) = \frac{1}{16} \times \begin{vmatrix} 1 & 3 \\ \binom{2}{(20),(01)} & \binom{2}{(11),(01)}+\binom{2}{(02),(01)} \end{vmatrix}$$

$$= \frac{1}{16} \times \begin{vmatrix} 1 & 3 \\ 0 & 4 \end{vmatrix}$$

$$= \frac{1}{8}.$$

In like fashion, the remaining coefficients are

$$C_0'\left((20),\begin{pmatrix}00\\11\\00\end{pmatrix}\right)=\frac{1}{8}$$

$$C_0'\left((20),\begin{pmatrix}00\\20\\00\end{pmatrix}\right)=-\frac{3}{16}$$

and

$$C_0'\left((20),\begin{pmatrix}00\\02\\00\end{pmatrix}\right)=\frac{1}{16}.$$

With the required coefficients provided above the value of $-\left|\overline{P}_{\overline{m}_A}' - \overline{I}'\right|$ for $\overline{m}_A = (20)$ can be expressed (by Eq. 9.26) as

$$-|\overline{P}_{(20)}' - \overline{I}'| = -\frac{1}{4} - \frac{1}{4}(2P(0 \mid (11)) - 1) + \frac{1}{4}(2P(1 \mid (11)) - 1)$$

$$+\frac{1}{8}(2P(0 \mid (11)) - 1)(2P(1 \mid (11)) - 1) \qquad \text{Eq. 9.29}$$

$$-\frac{3}{16}(2P(0 \mid (11)) - 1)^2 + \frac{1}{16}(2P(1 \mid (11)) - 1)^2.$$

Then, since $P(0 \mid 11) + P(1 \mid 11) = 1 \Rightarrow (2P(1 \mid 11) - 1) = -(2P(0 \mid 11) - 1)$, Eq. 9.29 simplifies to

$$-|\overline{P}_{(20)}' - \overline{I}'| = -\frac{1}{4} - \frac{1}{2}(2P(0 \mid (11)) - 1) - \frac{1}{4}(2P(0 \mid (11)) - 1)^2$$

$$= -\frac{1}{4}[1 + (2P(0 \mid (11)) - 1)]^2 \qquad \text{Eq. 9.30}$$

$$= -P(0 \mid 11)^2.$$

From the symmetry inherent in the problem, it follows that the $\overline{m}_A = (02)$ counterpart of Eq. 9.30 is

$$-|\overline{P}_{(02)}' - \overline{I}'| = -P(1 \mid 11)^2, \qquad \text{Eq. 9.31}$$

and employing Eq. 9.30-31 with Proposition 7.5 yields (for the two-operator case)

$$q_0(20) = \frac{P_1(0 \mid 11)^2}{P_1(0 \mid 11)^2 + P_1(1 \mid 11)^2}$$

and Eq. 9.32

$$q_0(02) = \frac{P_1(1 \mid 11)^2}{P_1(0 \mid 11)^2 + P_1(1 \mid 11)^2}.$$

Then, substituting Eq. 4.2 in Eq. 9.32 yields

$$q_0(20) = \frac{R(0)^2}{R(0)^2 + R(1)^2}$$

and Eq. 9.33

$$q_0(02) = \frac{R(1)^2}{R(0)^2 + R(1)^2}.$$

The limit for the nonabsorbing state $\overline{m} = (11)$ is known to be zero by Proposition 7.5. An identical result to Eq. 9.33 obtains for the three-operator case because for the one-bit problem, crossover is nullified and $P_2'(i \mid \overline{n}) = P_1(i \mid \overline{n})$ (see Eq. 4.22).

Additional insight into the behavior of the limiting stationary distribution is obtainable by examining the one-bit problem with population size 3. These parameters $(L = 1, M = 3)$ leave S and N unaltered but change the other state space related sets and parameters to $S' = \{(30), (21), (12), (03)\}$, $N' = 4$, $S_A' = \{(30), (03)\}$ and $S' - S_A' = \{(21), (12)\}$. By retracing the previous development (the $M = 2$ case) with $\bar{r}$ limited as indicated by these state space sets, results analogous to Eq. 9.32 and 9.33 are obtained. Thus, the M=3 counterpart of Eq. 9.32 is

$$q_0(30) = \frac{\left\{ \begin{array}{l} P_1(0 \mid 21)^3 [P_1(0 \mid 12)^3 + 3P_1(0 \mid 12)^2 P_1(1 \mid 12)] + \\ \quad P_1(0 \mid 12)^3 [P_1(1 \mid 21)^3 + 3P_1(0 \mid 21) P_1(1 \mid 21)^2] \end{array} \right\}}{D}$$

and Eq. 9.34

$$q_0(03) = \frac{\left\{ \begin{array}{l} P_1(1 \mid 21)^3 [P_1(1 \mid 12)^3 + 3P_1(1 \mid 12)^2 P_1(0 \mid 12)] + \\ \quad P_1(1 \mid 12)^3 [P_1(0 \mid 21)^3 + 3P_1(1 \mid 21) P_1(0 \mid 21)^2] \end{array} \right\}}{D}$$

where

$$D = P_1(0 \mid 21)^3 [P_1(0 \mid 12)^3 + 3P_1(0 \mid 12)^2 P_1(1 \mid 12)] +$$

$$P_1(0 \mid 12)^3 [P_1(1 \mid 21)^3 + 3P_1(0 \mid 21) P_1(1 \mid 21)^2] +$$

$$P_1(1 \mid 12)^3 [P_1(1 \mid 21)^3 + 3P_1(1 \mid 21)^2 P_1(0 \mid 21)] +$$

$$P_1(1 \mid 21)^3 [P_1(0 \mid 12)^3 + 3P_1(1 \mid 12) P_1(0 \mid 12)^2].$$

The Eq. 9.33 counterpart is

$$q_0(30) = \frac{[2R(0)]^3 [R(0)^3 + 6R(0)^2 R(1)] + R(0)^3 [R(1)^3 + 6R(0)R(1)^2]}{D'}$$

and Eq. 9.35

$$q_0(03) = \frac{[2R(1)]^3 [R(1)^3 + 6R(1)^2 R(0)] + R(1)^3 [R(0)^3 + 6R(1)R(0)^2]}{D'}$$

where

$$D' = [2R(0)]^3 [R(0)^3 + 6R(0)^2 R(1)] + R(0)^3 [R(1)^3 + 6R(0)R(1)^2]$$

$$+[2R(1)]^3 [R(1)^3 + 6R(1)^2 R(0)] + R(1)^3 [R(0)^3 + 6R(1)R(0)^2].$$

Again, the three-operator case yields an identical result.

These examples suggest two very significant conjectural features of the limiting stationary distribution behavior. First, only order 2 monomial terms survive in the determinant expansions of the M=2 case and only order 6 terms survive for M=3. These facts lead to the supposition that in general, only order Mx(N'-N) terms survive. In the M=2 case, Mx(N'-N) = 2x(3-2) = 2 while for M=3, Mx(N'-N) = 3x(4-2) = 6. If this supposition is correct, then the polynomial forms required for evaluating the stationary distribution zero mutation probability limit by Propositions 7.5 and 7.6 are homogeneous order Mx(N'-N) polynomials in the $P(i \mid \bar{n})$'s. Presumably, the corresponding property (i.e. homogeneous order Mx(N'-1) order polynomial forms) applies to the general case represented by Propositions 6.7 and 6.8.

A second conjecture concerns the limiting distribution behavior as a function of the parameter M. The computed limiting distribution entropy results displayed in Section 5.5 suggest that the limiting distribution is dominated by optimal solutions for M sufficiently large. That supposition is supported by the results in Eq. 9.33 and 9.35. In the M=2 case, it follows immediately from Eq. 9.33 that $q_0(02)/q_0(20) = [R(1)/R(0)]^2$. For M=3 and R(1) < R(0) it is straightforward to show that a corresponding bounding relationship exists, i.e. $q_0(03)/q_0(30) < [R(1)/R(0)]^3$. This suggests that the ratio of the probabilities of the uniform population states corresponding to i and j with R(i) < R(j) behaves at or better than

$$[R(i)/R(j)]^M \rightarrow 0 \qquad \text{Eq. 9.36}$$

for M sufficiently large. If this supposition is indeed correct, then the desired limiting distribution behavior for the two-operator simple genetic algorithm (i.e. probability zero for sub-optimal solutions) can be approached as closely as required by selecting M sufficiently large.

The corresponding general case (i.e. L>1) three-operator counterparts of Eq. 9.32 and 9.34 are expressed in terms of the $P_2(i \mid \overline{n})$' array (Eq. 4.22). Thus, the numerator polynomial counterparts of Eq. 9.33 and 9.35 are expressed in terms of complex polynomial functions of the reward function values, and consequently it may be that no general case three-operator counterpart of Eq. 9.36 exists. (It is noted that the design of the reward functions employed in Section 5, in which only length 0-2 schema dependence is incorporated, tends to minimize crossover disruption, which may account for the progression toward optimality indicated by the three-operator results recorded in Figures 5-7 through 5-18). The simulated annealing global optimality may thus extrapolate onto the simple genetic algorithm only in the $p_c \rightarrow 0$ and $M \rightarrow \infty$ limiting sense.

9.5 Extending the Stationary Distribution Representation

Eq. 9.26 and 9.27 represent an exact expression of the value of the determinant $-|\overline{P}_{\overline{m}} - \overline{I}| = |\overline{P} - \overline{I}| - |\overline{P}_m - \overline{I}|$, and with Propositions 6.7 and 6.8 constitute an exact representation of the components of the stationary distribution of the two and three-operator

algorithms. Section 9.4 extends those results to the determinants $-\left|\bar{P}_{\bar{m}_A}' - \bar{I}'\right| = |\bar{P}(\bar{m}_A)' - \bar{I}'| - \left|\bar{P}_{\bar{m}_A}' - \bar{I}'\right|$ whose $\alpha \to 0^+$ values are required for use in Propositions 7.5-6. The utility of these representations depends upon the ability to extract useful relationships between the $C'(\bar{m}, \bar{r})$'s from the general form represented by Eq. 9.27 and Eq. 9.28. The following paragraphs examine the combinatorial determinants in the general forms provided by Eq. 9.27 and Eq. 9.28 and deduce some of the key relationships. The purpose of this effort is to provide a foundation for extending the stationary distribution representation methodology developed in Sections 9.2-9.4.

First, if the enabling condition for Eq. 9.22 is satisfied (i.e. $M \le N$), then every element in the combinatorial determinant of Eq. 9.27 is either zero or it is the combinatorial determinant corresponding to the order zero coefficient for some state in S'. Thus, every coefficient of the form represented by Eq. 9.27 can be written as sums and products of order zero coefficients. An analogous conclusion applies to Eq. 9.28.

Second, it is clear from Eq. 9.27 that nonzero order differentiation of any two or more rows of $-|\bar{P}_{\bar{m}} - \bar{I}| = |\bar{P} - \bar{I}| - |\bar{P}_{\bar{m}} - \bar{I}|$ in an identical pattern (e.g. $\bar{r}' : \bar{0} \neq \bar{r}_{\bar{n}_1} = \bar{r}_{\bar{n}_2}$ for $\bar{n}_1 \neq \bar{n}_2$) introduces identical rows into the combinatorial determinant, and thus makes $C'(\bar{m}, \bar{r}') \equiv 0$. Consequently, no monomial terms corresponding to any $\bar{r}'$ with identical nonzero rows survive in the expansion of $-|\bar{P}_{\bar{m}} - \bar{I}| = |\bar{P} - \bar{I}| - |\bar{P}_{\bar{m}} - \bar{I}|$. An identical conclusion applies to the coefficients of $-\left|\bar{P}_{\bar{m}_A}' - \bar{I}'\right| = |\bar{P}(\bar{m}_A)' - \bar{I}'| - \left|\bar{P}_{\bar{m}_A}' - \bar{I}'\right|$, of which Eq. 9.28 is an exemplar.

A very important class of coefficient identities derives from transpositions of nonzero rows and columns of the differentiation order array. The resulting identities are very closely connected to the algebra of symmetric and alternating polynomials, and to an associated determinant concept called alternants, of which Vandermonde determinants are a special case. Appendix C is provided to support the following paragraphs.

From the form of the combinatorial determinants in Eq. 9.27 and Eq. 9.28, it is clear that exchanging any two of the k rows indexed by row indices $\bar{n} \in K$ is equivalent to

exchanging the corresponding nonzero rows of $\bar{r}$. If $\bar{r}$' is derived from $\bar{r}$ by such a row transposition, then it follows that

$$C'(\bar{m},\bar{r}') = -C'(\bar{m},\bar{r}).$$

Thus, $C'(\bar{m},\bar{r})$ establishes the value (to within a sign alternation) of the coefficients of k! distinct monomial terms in the expansion of $-|\bar{P}_{\bar{m}} - \bar{I}| = |\bar{P} - \bar{I}| - |\bar{P}_{\bar{m}} - \bar{I}|$. An identical result applies to $C'_0(\bar{m},\bar{r})$ and the expansion of $-\left|\bar{P}_{\bar{m}_A}' - \bar{I}'\right| = |\bar{P}(\bar{m}_A)' - \bar{I}'| - \left|\bar{P}_{\bar{m}_A}' - \bar{I}'\right|$.

The collection of monomial terms corresponding to this coefficient identity can be written as the product of $C'(\bar{m},\bar{r})$ (or of $C'_0(\bar{m},\bar{r})$) and a polynomial function of the form defined in Eq. C.12 of Appendix C. That is, the collection of terms is a quasi-alternating polynomial function in the array of variables $(2^L P(i \mid \bar{n}) - 1)$.

In addition to the preceding result, the following identity applies to $C'(\bar{m},\bar{r})$ and the expansion of $-|\bar{P}_{\bar{m}} - \bar{I}| = |\bar{P} - \bar{I}| - |\bar{P}_{\bar{m}} - \bar{I}|$. For any $\bar{n} \in K$, transposition of columns $\bar{m}$ and $\bar{n}$ in the combinatorial determinant of Eq. 9.27 is equivalent to representing the value of $C'(\bar{n},\bar{r}')$ where $\bar{r}'$ is derived from $\bar{r}$ by exchanging $\bar{r}_{\bar{n}}$ with $\bar{r}_{\bar{m}} = \bar{0}$. That is

$$\bar{n} \in K \Rightarrow C'(\bar{n},\bar{r}') = -C'(\bar{m},\bar{r}).$$

Thus, the identical quasi-alternating function, evaluated in the new set of variables generated by replacing each $P(i \mid \bar{n})$ with the corresponding $P(i \mid \bar{m})$, is included in the expansion of $-|\bar{P}_{\bar{n}} - \bar{I}| = |\bar{P} - \bar{I}| - |\bar{P}_{\bar{n}} - \bar{I}|$. Collectively, these results account for $(k+1)!$ of the coefficients required for representation of the stationary distribution.

Another class of coefficient identities derives from transpositions of the columns of $\bar{r}$ (i.e. transpositions of $i,j \in S$). Let $\bar{m}$' be derived from $\bar{m}$ by setting $m(j)' = m(i)$, $m(i)' = m(j)$, $\bar{n}_1$' from $\bar{n}_1$ by setting $n_1(j)' = n_1(i)$, $n_1(i)' = n_1(j)$, etc. Then, if $\bar{r}$' is derived from $\bar{r}$ by transposition of rows $\bar{m}$ with $\bar{m}$', $\bar{n}_1$ with $\bar{n}_1$', etc. followed by transposition of columns i and j, it follows from Eq. 9.27 that

$$C'(\bar{m}',\bar{r}') = C'(\bar{m},\bar{r}).$$

An identical result applies to $C'_0(\bar{m}_A',\bar{r}')$.

The number of distinct coefficients whose values are generated in this fashion from $C'(\overline{m},\bar{r})$ depends upon both the number and form of the nonzero columns in $\bar{r}$. If the number of nonzero columns is p, then exchanging any of the p nonzero columns with any of the N - p zero columns generates the coefficient of a distinct monomial term. Exchanging a nonzero column with another nonzero column having a different column sum also generates a distinct coefficient. However, exchanging a nonzero column with another nonzero column having identical column sum may or may not generate a distinct coefficient, depending upon the distribution of the nonzero entries in the two columns, because it is possible for the transformation described above to translate one column into the other. A lower bound on the number of distinct coefficients thus generated is

$$\binom{N}{p}.$$

The collection of monomial terms corresponding to this coefficient identity can be written as the product of $C'(\overline{m},\bar{r})$ (or of $C'_0(\overline{m},\bar{r})$) and a polynomial function of the form defined in Eq. C.10 of Appendix C. That is, the collection of terms is a quasi-symmetric polynomial function in the array of variables $(2^L P(i \mid \overline{n}) - 1)$.

These coefficient identities and their connection to the quasi-symmetric and quasi-alternating polynomials of Appendix C offer a promising mechanism for extending the stationary distribution representation work begun here. Examination of the general form $(2^L P(i \mid \overline{n}) - 1)$ reveals that it is zero mean in the sense that

$$\sum_{i \in S} (2^L P(i \mid \overline{n}) - 1) = 0.$$

This property, along with the common form of the elements in the conditional probability array $[P(i \mid \overline{n})]$, suggests that the symmetric and alternating polynomial forms required for evaluation of Propositions 6.7-6.8 or 6.5-6.6 may admit to large scale simplifications, and ultimately yield a tractable, explicit closed form expression for the stationary distribution components.

SECTION 10

CONCLUSIONS AND FUTURE DIRECTION

10.1 Summary

This dissertation reports an effort to establish an analytical framework for the simple genetic algorithm, based upon the asymptotic probability distribution of the generated solution sequences. The mechanism employed herein is extrapolation of the extensive existing theoretical foundation of the simulated annealing algorithm onto the genetic algorithm. That foundation is based upon the asymptotic behavior of a nonstationary Markov chain simulated annealing algorithm model. The simulated annealing literature is reviewed in Section 2, with particular emphasis on the methodology employed to develop the key theoretical results. Those results include a demonstration that provided a lower bound of the form K/log(k) on the algorithm parameter corresponding to absolute temperature is observed, the asymptotic probability distribution over the algorithm state space is zero for all states corresponding to sub-optimal solutions. Thus, the simulated annealing algorithm obtains (asymptotically) a globally optimal solution.

The genetic algorithm literature is reviewed in Section 3. The significant conclusion of that section is that while certain important theoretical results exist, notably the so called schema theorem and some work on a problem construct referred to as the minimal deceptive problem, no genetic algorithm model or accompanying convergence theory comparable in scope to that of simulated annealing exists in the literature. The fundamental purpose of the work described herein is to provide such an analytical framework by extrapolating the known simulated annealing theory onto the genetic algorithm.

An essential first step toward that goal is development of a nonstationary Markov chain algorithm model for the genetic algorithm. That task is accomplished in Section 4.

The product of that effort is a very general nonstationary Markov chain model for variants of the algorithm incorporating combinations of the three fundamental genetic algorithm operators. The model is tailored to resemble the model employed in the analysis of the simulated annealing algorithm convergence behavior, with the mutation probability algorithm parameter playing a role analogous to absolute temperature in simulated annealing.

Additionally, some salient features of the model state behavior are pointed out in Section 4. In particular, the one-operator (reproduction only) simple genetic algorithm is shown to possess exactly 2^L absorbing states, one for each possible uniform population, while the two-operator (reproduction/mutation) and three-operator (reproduction/mutation/crossover) variants possess a unique stationary distribution. The expected value of the absorption time for the one-operator algorithm is finite and an upper bound is provided by Eq. 4.8. The probability distribution of the final solution state produced by the one-operator simple genetic algorithm depends upon the initial state, $\overline{m}_0$.

The inclusion of the mutation operator is shown in Section 4 to provide a significant additional dimension to the state behavior of the time-homogeneous (stationary) two and three-operator variants of the algorithm, the existence of a unique stationary distribution. The significance of the unique stationary distribution is that the asymptotic state behavior is independent of the starting state. It is completely determined by the objective function and the algorithm parameters.

In Section 5, the genetic algorithm model is employed to generate some computer simulation results. Specifically, a combinatorial interpretation of the model state space is explored numerically in Section 5.2 and the limiting stationary distribution of the three operator algorithm is approximated for a variety of algorithm parameter sets in Section 5.5. A very significant feature of the limiting stationary distribution is suggested by the Section 5.5 results and later verified theoretically (in Section 7). It is that the limiting two and three-operator algorithm stationary distribution behavior necessary for extrapolating

the simulated annealing asymptotic global optimality result does not follow. The limiting distribution is nonzero for all states corresponding to uniform populations (one-operator absorbing states), including those representing sub-optimal solutions. This complication precludes an exact extrapolation of the simulated annealing convergence theory onto the simple genetic algorithm. The Section 5 results do however reinforce the intuitive notion that increasing the algorithm population size parameter biases the limiting distribution towards the desired limiting behavior.

Section 6 employs the Perron-Frobenius Theorem (which is summarized in Appendix B) to formulate the time-homogeneous two and three-operator algorithm unique stationary distribution existence argument into a system of equations whose solution is the stationary distribution components. The solution is formulated in terms of Cramer's Rule, and is not explicitly solved, however the Section 6 results provide a remarkable degree of insight into the form of the solution and its behavior with respect to the algorithm parameters. Those results provide the foundation for the remaining sections.

The unique stationary distribution existence argument for the stationary two and three-operator algorithm variants only applies when the mutation probability parameter is strictly greater than zero. A one-operator (zero mutation probability) stationary distribution exists but as demonstrated in Section 4.3.1 it is not unique. A very important requirement for extrapolation of the simulated annealing convergence theory onto the simple genetic algorithm is existence of a zero mutation probability limit for the stationary distribution. Section 7 is devoted to resolving that question affirmatively. It is based upon the results developed in Section 6 and it also verifies the Section 5.5 observation concerning the nonzero limit for all states corresponding to uniform populations.

A very significant theoretical contribution of this work is developed in Section 8. It is a monotonic mutation probability bound sufficient to guarantee strong ergodicity of the nonstationary two and three-operator simple genetic algorithm Markov chains. The parameter bound is analogous to the simulated annealing temperature schedule bound.

The bound is asserted in Proposition 8.1, and its form (i.e. $k^{-\gamma}$) is asymptotically superior to the K/log(k) bound associated with the simulated annealing algorithm. It is very noteworthy that the same bound applies both to the two and three-operator algorithm variants. At least in terms of the Section 8 bound, the crossover operator does not expedite convergence.

All of the results developed in Sections 7 and 8 are obtained without explicitly solving the stationary distribution system. Section 9 attacks the problem of explicit solution. It is a very extensive and somewhat tedious development. The product of that work is an expression for the general term in a multivariate Taylor's series expansion of the determinant form required for explicit solution of the stationary distribution equations. The results are expressed in Eq. 9.26 and 9.27 for the general nonzero mutation probability case, augmented by Eq. 9.28 for the zero mutation probability limit. These results stop short of a useable answer but they do provide some insight into the nature of the solution. Further, Section 9.5 provides some intriguing ideas for extending the work started in Section 9.

The attempt to extrapolate the simulated annealing convergence theory onto the genetic algorithm fails in the sense that the zero mutation probability stationary distribution limits of the two and three-operator simple genetic algorithm variants do not satisfy the required form for extrapolation of the simulated annealing global optimality result. However, evidence is provided which suggests that for the two-operator algorithm variant, the required behavior can be approached by increasing the population size parameter (Eq. 9.36). The question is more complicated for the three-operator case, and as pointed out in Section 9.4, implementation of crossover with nonzero p_c may indeed preclude convergence to global optimality even in the infinite population size limiting sense of Eq. 9.36.

The latter observation concerning crossover, along with the equivalence of the mutation probability sequence bounds for the two and three-operator cases noted

previously, poses some significant questions concerning the role of the crossover operator. Indeed, from the results developed herein, it is not clear that any desirable effect on the asymptotic algorithm behavior obtains from application of the crossover operator, though it may have a desirable effect in expediting convergence in real (finite time) applications. The resolution of these questions, along with a host of other applications questions such as optimum population size, mutation and crossover probability parameter selection, number of iterations required to achieve acceptable results, etc. require further progress on the stationary distribution representation task begun in Section 9.

10.2 Contributions of the Research

The research reported herein establishes a framework for modeling the genetic algorithm in terms of the asymptotic probability distribution of the solution sequences which it produces. Specific significant accomplishments include the following:

(1) A very general nonstationary Markov Chain model of one, two and three-operator variants of the genetic algorithm, and a framework for analysis of the operators based upon their impact on the state space of the Markov chain

(2) Demonstration of the existence of a unique stationary distribution for the time-homogeneous (stationary) two and three-operator algorithm variants

(3) A stationary distribution solution in terms of the characteristic polynomials of matrices derived from the state transition matrix

(4) Demonstration of the existence of a zero mutation probability stationary distribution limit for the time-homogeneous two and three-operator algorithms

(5) A mutation probability schedule bound (analogous to the annealing schedule bound of simulated annealing) sufficient for the nonstationary two and three-operator genetic algorithm variants to achieve the limiting distribution

(6) A methodology for representing the two and three-operator stationary distribution components at all consistent values of mutation probability (including the zero mutation probability limit), and a proposed approach for extending that methodology to produce an explicit result.

10.3 Future Direction

In order to achieve the stated goal of this work, a complete analytical framework for the simple genetic algorithm, additional progress must be made on the stationary distribution solution effort begun in Section 9. The coefficient relationships noted in Section 9.5, especially the coefficient identities which attend transpositions of rows and columns in the differentiation order array and their connection with the quasi-symmetric and quasi-alternating polynomial notions presented in Appendix C, provide a foundation for proceeding with this effort. An explicit representation of the functional form of the stationary distribution, reduced to a rational function expression in the algorithm parameters and objective function, would provide a very valuable theoretical tool for use in the analysis of genetic algorithm performance, and is the ultimate goal. However, even if explicit solution is not attainable, it may prove possible to deduce very useful bounds on the stationary distribution components from continuation of the Section 9 development.

A second promising area for continuation of this work concerns the mutation probability parameter sequence bound provided in Section 8. It is based on very simple lower bounds (Eq. 4.21 and 4.31) which exist for the conditional probabilities which compose the state transition matrix, and it only employs the one-step transition matrix in the $(1 - \tau_1(\bar{P}))$ sequence employed to establish weak ergodicity (Section 8.2). Some preliminary work not reported in Section 8 suggests that employing two-step transition matrices in summing the $(1 - \tau_1(\bar{P}))$ sequence may allow a refinement of the bound to something of the form k^{-1}. It also appears from that preliminary work that the same bound applies for both the two and three-operator algorithm variants.

APPENDIX A

DISCRETE TIME FINITE STATE MARKOV CHAINS

A.1 Introduction

The following paragraphs establish some definitions and theorems on discrete time finite state Markov chains and related stochastic matrix concepts. These results fall into three main categories, (1) elementary definitions, (2) definitions and theorems concerning the state space and asymptotic behavior of time-homogeneous (stationary) Markov chains and (3) some more advanced ergodicity definitions and theorems necessary for the analysis of the asymptotic behavior of inhomogeneous Markov chains. These results are presented without proof or elaboration but the foundation required for the more elementary of them can be obtained from [Cinl75, IsMa76] or many other references on Markov chains. The ergodicity related results can be found in [IsMa76, Sene81].

Although some of the results discussed here apply to continuous time and/or denumerably infinite state space Markov chains as well, the intention is to restrict consideration to the discrete time finite state case. All references herein to Markov chains are understood to mean discrete time finite state Markov chains.

In the following, let $K = \{0, 1, 2, \cdots\}$ be the set of nonnegative integers, let $X = \{X_k : k \in K\}$ be a discrete time (i.e. discrete sequence index) stochastic process with finite cardinality state space E, and let $i,j \in E$.

A.2 Elementary Definitions

Definition A1: If $\forall i,j \in E$ and every $k \in K$, it follows that

$$\Pr\{X_{k+1} = j : X_0 = i_0, X_1 = i_1, \cdots, X_k = i\} = \Pr\{X_{k+1} = j : X_k = i\},$$

then X is a Markov chain.

Definition A2: Any row vector $\bar{q}^T = [q(i)]$, $i \in E$ satisfying the conditions

$$(1) \quad \forall i \in E : q(i) \geq 0$$

$$(2) \quad \bar{q}^T \bar{1} = \sum_{i \in E} q(i) = 1$$

is called a probability vector.

Definition A3: Any square matrix $\bar{P}$ whose rows are all composed of probability vectors is called a stochastic matrix, or sometimes more explicitly a row stochastic matrix. The row sum constraint on a row stochastic matrix can be written as $\overline{P1} = \bar{1}$, so 1 is an eigenvalue of every stochastic matrix.

Definition A4: The stochastic matrix

$$\bar{P}_k = [p_k(i,j)] = [\Pr\{X_{k+1} = j \mid X_k = i\}]$$

is the one step transition probability matrix or state transition matrix of the Markov chain X. If the probability vectors $\bar{q}_k$ and $\bar{q}_{k+1}$ are respectively the probability distributions of X_k and X_{k+1}, then

$$\bar{q}_{k+1}^T = \bar{q}_k^T \bar{P}_k.$$

Similarly, the stochastic matrix

$$\bar{P}_{mk} = [p_{mk}(i,j)] = [\Pr\{X_k = j \mid X_m = i\}] = \bar{P}_m \bar{P}_{m+1} \cdots \bar{P}_{k-1} = \prod_{l=m}^{k-1} \bar{P}_l$$

where $k = m + n$, $m, n \in K$, $n > 0$ is the n-step transition probability matrix of X, and

$$\bar{q}_k^T = \bar{q}_m^T \bar{P}_{mk} = \bar{q}_m^T \prod_{l=m}^{k-1} \bar{P}_l.$$

Definition A5: Let $\bar{P}_k$ be the state transition matrix of the Markov chain X at time (sequence index) k. Then, X is time-homogeneous if and only if $\forall k \in K$ it follows that $\bar{P}_k = \bar{P}$ where $\bar{P}$ is a constant state transition matrix.

A.3 Time-Homogeneous Markov Chains

The time-homogeneous Markov chain X is completely specified by its initial probability distribution, $\bar{q}_0$, and state transition matrix, $\bar{P}$. The probability distribution of X_k, $k \geq 1$ is given by

$$\bar{q}_k^T = \bar{q}_0^T \bar{P}^k.$$

The following definitions and theorems concern the asymptotic behavior of the chain and some conditions on the state space which make the asymptotic behavior independent of $\bar{q}_0$. In the following, let the $i,j \in E$ element of $\bar{P}^k$ be denoted by $p^{(k)}(i,j)$.

Definition A6: A subset E_0 of the state space E of the Markov chain X is called closed if $\forall i \in E_0$, $\forall j \in E - E_0$ it follows that $p(i,j) = 0$. If the closed set E_0 contains the single state i, so that $p(i,i) = 1$, then the state i is called an absorbing state.

Definition A7: A Markov chain is called irreducible if there exists no nonempty closed subset of its state space E other than E itself.

Definition A8: The states i and j are said to intercommunicate if

$$\exists k_i, k_j \in K \ni p^{(k_i)}(i,j) > 0 \text{ and } p^{(k_j)}(j,i) > 0.$$

Theorem A1: A Markov chain is irreducible if and only if all pairs of states intercommunicate.

Definition A9: State $i \in E$ of the Markov chain X has period d if the following two conditions hold:

(1) $p^{(k)}(i,i) = 0$ unless $k = md$ for some positive integer m and

(2) d is the largest integer with property (1).

If $d = 1$, state i is called aperiodic. The Markov chain X is aperiodic if and only if $\forall i \in E$ are aperiodic.

Theorem A2: If X is irreducible and if $\exists i \in E \ni p(i,i) > 0$, then X is aperiodic.

Definition A10: Any probability vector $\bar{q}$ over the state space of the time homogeneous Markov chain X and satisfying

$$\bar{q}^T\bar{P} = \bar{q}^T,$$

is called a stationary distribution of X. It is not necessarily unique.

Theorem A3: If the Markov chain X is time-homogeneous, irreducible, aperiodic and has a finite state space, then a stationary distribution exists for X. Further, the stationary distribution is unique and is determined by

$$\bar{q}^T\bar{P} = \bar{q}^T$$

and

$$\bar{q}^T\bar{1} = 1.$$

Theorem A4: If the time-homogeneous Markov chain X possesses a unique stationary distribution, $\bar{q}$, then for every probability vector $\bar{x}$ with compatible dimensions, it follows that

$$\lim_{k \to \infty} \bar{x}^T\bar{P}^k = \bar{q}^T.$$

A.4 Inhomogeneous Markov Chains

Complete specification of the inhomogeneous Markov chain X requires its initial probability distribution, $\bar{q}_0$, and the infinite sequence of state transition matrices, $\{\bar{P}_k\}$, $k \geq 0$. The probability distribution of X_k, $k \geq 1$ is given by

$$\bar{q}_k^T = \bar{q}_0^T \prod_{n=0}^{k-1} \bar{P}_n.$$

If the chain is asymptotically independent of $\bar{q}_0$, then it is said to be ergodic. Two classes of ergodicity must be distinguished. The following definitions and theorems elaborate.

Definition A11: The inhomogeneous Markov chain X is weakly ergodic if

$\forall i, j, l \in E, \forall m \in K$

$$\lim_{k \to \infty} (p_{mk}(i,l) - p_{mk}(j,l)) = 0.$$

Weak ergodicity does not require that either $\lim_{k \to \infty} p_{mk}(i,l)$ or $\lim_{k \to \infty} p_{mk}(j,l)$ exist.

Definition A12: Any scalar function $\tau(\cdot)$, continuous on the set of nxn stochastic matrices $\bar{P}$ and satisfying $0 \leq \tau(\bar{P}) \leq 1$ is called a coefficient of ergodicity. If in addition

$$\tau(\bar{P}) = 0 \Leftrightarrow \bar{P} = \bar{1}\bar{q}^T$$

where $\bar{q}$ is any probability vector with compatible dimensions (i.e. when all rows of $\bar{P}$ are identical probability vectors), then τ is said to be proper. Weak ergodicity is equivalent to

$$\tau(\bar{P}_{mk}) \to 0, k \to \infty, m \geq 0$$

where τ is a proper coefficient of ergodicity.

Theorem A5: Let $\bar{P}$ be a nxn stochastic matrix and let

$$\tau_1(\bar{P}) = 1 - \min_{i,j} \sum_{k=1}^{n} \min(p(i,k), p(j,k)).$$

Then, $\tau_1(\bar{P})$ is a proper coefficient of ergodicity.

Theorem A6: The inhomogeneous Markov chain X is weakly ergodic if and only if there exists a strictly increasing sequence of positive numbers $\{k_l\}$, $l \in K$ such that

$$\sum_{l=0}^{\infty}\left(1-\tau_1\left(\overline{P}_{k_l,k_{l+1}}\right)\right)=\infty.$$

Definition A13: The inhomogeneous Markov chain X is strongly ergodic if there exists a probability vector $\overline{q}$ satisfying

$$\forall i,j \in E, \forall m \in K: \lim_{k\to\infty} p_{mk}(i,j) = q(j).$$

Thus, strong ergodicity implies convergence in distribution. The unique vector $\overline{q}$ is analogous to its time-homogeneous chain counterpart in Theorem A4.

Theorem A7: The inhomogeneous Markov chain X is strongly ergodic if it is weakly ergodic and if for each transition matrix, $\overline{P}_k$, of X there exists a left eigenvector $\overline{q}_k$ corresponding to eigenvalue 1, $\overline{q}_k$ is a probability vector, and

$$\sum_{k=0}^{\infty}\sum_{i\in E}|q_k(i)-q_{k+1}(i)| < \infty.$$

Further, if $\overline{q} = \lim_{k\to\infty} \overline{q}_k$, then $\overline{q}$ is the unique vector in Definition A13.

APPENDIX B

THE PERRON-FROBENIUS THEOREM AND STOCHASTIC MATRICES

B.1 Introduction

A matrix possessing the property that all of its components are nonnegative is referred to as a nonnegative matrix. For the matrix $\overline{T}$, this condition is indicated by $\overline{T} \geq \overline{0}$. The case in which all components of $\overline{T}$ are strictly positive is indicated by $\overline{T} > \overline{0}$. This notation extends in the obvious manner to expressions such as $\overline{T} \geq \overline{B} \Leftrightarrow \overline{T} - \overline{B} \geq \overline{0}$ relating nonnegative matrices with compatible dimensions.

The definitions, theorems and corollary in Section B.2 below concern nonnegative matrices. They are the foundation for the Markov chain stationary distribution existence and representation theorem and related results summarized in Appendix A and employed in Sections 2, 4, 7 and 8. They are extracted from [Sene81], and are specialized in Section B.3 from the case of finite nonnegative matrices to the case of finite stochastic matrices. They are employed extensively in Sections 6 and 7.

B.2 The Perron-Frobenius Theorem and Ancillary Results for Primitive Matrices

Theorem B2 below is called the strong version of the Perron-Frobenius theorem. It applies to a class of nonnegative matrices referred to as primitive. A version of the theorem which applies to the wider class of irreducible nonnegative matrices is usually invoked for applications involving stochastic matrices, but the flexibility of the more general version is not required for the purposes herein. The connection of these results to those of Appendix A is provided by Theorem B1. It asserts that primitivity (Definition B1) is equivalent to the combination of irreducibility and aperiodicity, as defined in Appendix A.

Definition B1: A square nonnegative matrix, $\overline{T}$, is primitive if there exists a positive integer k such that $\overline{T}^k > \overline{0}$.

Theorem B1: If the $n \times n$ nonnegative matrix $\overline{T}$ is irreducible (Definition A7) and aperiodic (Definition A9), then $\overline{T}$ is primitive and conversely.

Theorem B2: Let $\overline{T}$ be an $n \times n$ nonnegative primitive matrix. Then there exists an eigenvalue r of $\overline{T}$ such that

(a) r is real, $r > 0$

(b) r has corresponding left and right eigenvectors with strictly positive components

(c) $r > |\lambda|$ for any eigenvalue $\lambda \neq r$

(d) the eigenvectors associated with r are unique to constant multiples

(e) If $\overline{0} \leq \overline{B} \leq \overline{T}$ and β is an eigenvalue of $\overline{B}$, then $|\beta| \leq r$. Moreover, $|\beta| = r$ implies $\overline{B} = \overline{T}$.

(f) r is a simple root of the characteristic polynomial of $\overline{T}$.

Definition B2: The eigenvalue r asserted in Theorem B2 is called the Perron-Frobenius eigenvalue of the nonnegative primitive matrix $\overline{T}$.

Corollary B1: Let $T_{i,j}$ be the components of a nonnegative primitive matrix $\overline{T}$ having Perron-Frobenius eigenvalue r. Then

$$\min_i \sum_j T_{i,j} \leq r \leq \max_i \sum_j T_{i,j}$$

with equality on either side implying equality throughout (i.e. r can only be equal to the maximal or minimal row sum if all row sums are equal). A similar proposition holds for column sums.

Theorem B3: Let $\overline{T}$ be an nxn nonnegative primitive matrix with Perron-Frobenius eigenvalue r, let $\overline{v}$ and $\overline{w}$ be strictly positive left and right eigenvectors respectively of $\overline{T}$ corresponding to r with $\overline{v}$ and $\overline{w}$ normed so that $\overline{v}^T\overline{w} = 1$, and let the $t \leq n$ distinct eigenvalues of $\overline{T}$ be ordered such that $r > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_t|$ with the additional condition that $|\lambda_2|$ has multiplicity m_2 equal to or greater than the multiplicity of any other eigenvalue λ_i for which $|\lambda_2| = |\lambda_i|$. It follows that

(a) if $\lambda_2 \neq 0$, then as $k \to \infty$

$$\overline{T}^k = r^k\overline{wv}^T + O(k^s|\lambda_2|^k)$$

elementwise, where $s = m_2 - 1$;

(b) if $\lambda_2 = 0$, then for $k \geq n - 1$

$$\overline{T}^k = r^k\overline{wv}^T.$$

B.3 The Perron-Frobenius Theory for Stochastic Matrices

A stochastic matrix (e.g. the state transition matrix of a Markov chain) is a special case of a square nonnegative matrix in which all row sums are equal to the constant 1. The following results specialize those of Section B.2 to the case of $\overline{T}$ an nxn stochastic primitive matrix, $\overline{P}$.

Theorem B4: Let $\overline{P}$ be an nxn stochastic primitive matrix. Then

(a) $r = 1$ is an eigenvalue of $\overline{P}$

(b) $r = 1$ has corresponding left and right eigenvectors with strictly positive components

(c) $r = 1 > |\lambda|$ for any eigenvalue $\lambda \neq r$

(d) the eigenvectors associated with $r = 1$ are unique to constant multiples

(e) If $\overline{0} \leq \overline{B} \leq \overline{P}$ and β is an eigenvalue of $\overline{B}$, then $|\beta| \leq r = 1$. Moreover, $|\beta| = r = 1$ implies $\overline{B} = \overline{P}$.

(f) $r = 1$ is a simple root of the characteristic polynomial of $\overline{P}$.

This theorem follows immediately from Theorem B2 by application of Corollary B1 with $\overline{T}$ a stochastic primitive matrix, $\overline{P}$. Among its consequences are the following.

Proposition B1: The right eigenvector asserted in Theorem B4(b) and (d) can be selected as the vector $\overline{1}$.

This result follows from the row sum constraint on nxn stochastic matrices, which can be expressed as $\overline{P1} = \overline{1}$. Thus, $\overline{1}$ is a right eigenvector, corresponding to eigenvalue 1, of every nxn stochastic matrix. Theorem B4 asserts that for finite primitive stochastic matrices, it is unique to within a nonzero scalar multiple.

Proposition B2: Let the vector $\overline{q}$ be the left eigenvector asserted in Theorem B4(b) and (d). Then, the additional constraint $\overline{q}^T\overline{1} = 1$ is consistent and makes $\overline{q}$ unique.

Since the left eigenvector asserted in Theorem B4(b) and (d) has strictly positive components, its inner product with the vector $\overline{1}$ is a strictly positive (nonzero) number. Consequently, that inner product can be used to normalize the eigenvector to produce a $\overline{q}$ which satisfies both requirements, and Proposition B2 follows.

Proposition B3: If $\overline{P}$ is an $n \times n$ stochastic primitive matrix, then

$$\lim_{k \to \infty} (\overline{P}^k) = \overline{1}\overline{q}^T$$

where $\overline{q}$ is the unique vector asserted in Proposition B2.

This result follows from Theorem B3 by specializing it to nxn stochastic primitive matrices via Theorem B4, Proposition B1 and Proposition B2. A very significant consequence is the following.

Proposition B4: If $\overline{P}$ is an $n \times n$ stochastic primitive matrix and $\overline{x}$ an arbitrary n-dimensional probability vector, then

$$\lim_{k \to \infty} (\overline{x}^T \overline{P}^k) = \overline{x}^T \overline{1} \overline{q}^T = \overline{q}^T$$

where $\overline{q}$ is the unique vector asserted in Proposition B2.

Definition B3: If the $n \times n$ stochastic primitive matrix $\overline{P}$ in Theorem B4 and Propositions B1-B4 is the state transition matrix of a time homogeneous Markov chain, then the unique vector $\overline{q}$ asserted in Proposition B2 is the stationary distribution of the Markov chain.

APPENDIX C

VANDERMONDE DETERMINANTS, SYMMETRIC AND ALTERNATING POLYNOMIALS

C.1 Introduction

An order n determinant whose i,j element is given by $\phi_j(x_i)$ for some set of n scalar functions ϕ_j and a companion set of n scalar variables x_i, i.e.

$$A_n(\bar{x}) = \begin{vmatrix} \phi_1(x_1) & \phi_2(x_1) & \cdots & \phi_n(x_1) \\ \phi_1(x_2) & \phi_2(x_2) & \cdots & \phi_n(x_2) \\ . & . & \cdots & . \\ . & . & \cdots & . \\ . & . & \cdots & . \\ \phi_1(x_n) & \phi_2(x_n) & \cdots & \phi_n(x_n) \end{vmatrix}, \qquad \text{Eq. C.1}$$

is called an alternant. The name derives from the fact that exchanging any pair of the variables in its argument list (e.g. x_p and x_q) affects the value of $A_n(\bar{x}) = A_n(x_1, x_2, \cdots, x_n)$ only by reversing its algebraic sign. This property is clear from Eq. C.1, because transposing the variables x_p and x_q in $A_n(\bar{x})$ amounts to exchanging the corresponding rows of the determinant, and from an elementary property of determinants, any such row exchange leaves the determinant value unchanged in magnitude but reversed in sign.

The state transition matrix of the Markov chain representing the simple genetic algorithm, as introduced in Section 4 of this paper, is a multivariate generalization of the matrix form underlying the alternant. The coefficient symmetries noted in Section 9.5 in connection with the stationary distribution representation development are a consequence. Section 10 proposes exploiting this connection in continuing the stationary distribution representation work begun in Section 9. This appendix provides some of the related background.

If the ϕ_j in Eq. C.1 are consecutive integer powers of their arguments, indexed from 0 through $n-1$, i.e.

$$A_n(\bar{x}) = |\bar{V}_n(\bar{x})| = \begin{vmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ . & . & . & \cdots & . \\ . & . & . & \cdots & . \\ . & . & . & \cdots & . \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} \end{vmatrix}, \qquad \text{Eq. C.2}$$

then the resulting special case alternant is known as a Vandermonde determinant. The values of a Vandermonde determinant and its minors are closely related to a class of polynomials in n variables referred to as the symmetric polynomials (and to a companion class of polynomials referred to as alternating polynomials). The distinguishing feature of the symmetric polynomials is invariance with respect to permutations of the argument list (e.g. $\psi(x_1, x_2, x_3) = x_1 + x_2 + x_3 = \psi(x_2, x_1, x_3)$). Alternating polynomials reverse sign with each transposition of variables.

Section C.2 below develops an expression for the value of the order n Vandermonde determinant in Eq. C.2. The evaluation method employs the determinant form and a polynomial remainder theorem due to Bezout. Section C.3 introduces formal definitions of symmetric and alternating polynomials and a fundamental theorem which associates them with Vandermonde determinants. Section C.4 generalizes the symmetric and alternating polynomial notions to the form required by the discussion in Section 9.5.

C.2 Evaluation of Vandermonde Determinants

The value of the order n Vandermonde determinant can be deduced from its form (Eq. C.2) and a polynomial remainder theorem due to Bezout. Let $\psi(\bar{x})$ be an arbitrary polynomial function in n variables (i.e. $\bar{x} = (x_1, x_2, \cdots, x_n)$) and let $\bar{x}_i^{(a)}$ be generated from $\bar{x}$ by replacing x_i with the value a. Then, the theorem states that if $\psi(\bar{x})$ is divided by the binomial $(x_i - a)$ the remainder is $\psi(\bar{x}_i^{(a)})$ [MoSt64]. That is

$$\psi(\bar{x}) = (x_i - a)\phi(\bar{x}) + \psi(\bar{x}_i^{(a)}).$$

If a is selected as $a = x_j$ for some $j \neq i$, then $\bar{x}_i^{(a)}$ contains the value x_j at two distinct index locations in its list (i.e. at i and j). Consequently, if $\psi(\bar{x})$ represents the value of the Vandermonde determinant in Eq. C.2 (i.e. $\psi(\bar{x}) = A_n(\bar{x}) = |V_n(\bar{x})|$), then the Vandermonde determinant represented by the polynomial function $\psi(\bar{x}_i^{(a)}) = A_n(\bar{x}_i^{(a)}) = |V_n(\bar{x}_i^{(a)})|$ contains two identical rows and hence is zero. In that case, the Bezout theorem reduces to

$$\psi(\bar{x}) = (x_i - x_j)\phi(\bar{x}).$$

Thus $(x_i - x_j)$ is a factor of $A_n(\bar{x}) = |V_n(\bar{x})|$.

This argument applies to each of the $n(n-1)/2$ distinct difference factors $(x_p - x_q)$. It follows that $A_n(\bar{x}) = |V_n(\bar{x})|$ can be written as

$$A_n(\bar{x}) = |V_n(\bar{x})| = \Phi(\bar{x}) \prod_{p=2, q<p}^{p=n} (x_p - x_q) \qquad \text{Eq. C.3}$$

where $\Phi(\bar{x})$ is a polynomial in $\bar{x} = (x_1, x_2, \cdots, x_n)$.

Every term in the polynomial expansion of the determinant in Eq. C.2 includes exactly one factor selected from each column. Thus, the order of every such term is

$$\sum_{k=0}^{n-1} k = n(n-1)/2.$$

The polynomial order of the product of difference factors in Eq. C.3 is also $n(n-1)/2$. Thus the polynomial $\Phi(\bar{x})$ in Eq. C.3 is exactly the zero order polynomial $\Phi(\bar{x}) = 1$, and consequently the value of the order n Vandermonde determinant in Eq. C.2 is

$$A_n(\bar{x}) = |V_n(\bar{x})| = \prod_{p=2, q<p}^{p=n} (x_p - x_q) \qquad \text{Eq. C.4}$$

It follows from Eq. C.4 that $A_n(\bar{x}) = |\overline{V}_n(\bar{x})|$ alternates sign with every exchange of two variables in its argument list. It also follows that $A_n(\bar{x}) = |\overline{V}_n(\bar{x})|$ is nonzero if and only if all of the x_k's are distinct.

C.3 Symmetric (and Alternating) Polynomials

A polynomial ψ in n variables having the property that

$$\psi(x_1, x_2, \cdots, x_n) = \psi(x_{\alpha_1}, x_{\alpha_2}, \cdots, x_{\alpha_n}) \qquad \text{Eq. C.5}$$

where $\bar{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_n)$ is a permutation of the integers $(1, 2, \cdots, n)$ is said to be symmetric with respect to the given permutation. If this property applies for all n! permutations of the x_k's, then ψ is a symmetric polynomial [MoSt64]. Some examples of symmetric polynomials are

$$\psi(x_1, x_2, x_3) = x_1 + x_2 + x_3$$

$$\psi(x_1, x_2, x_3) = x_1x_2 + x_1x_3 + x_2x_3$$

and

$$\psi(x_1, x_2, \cdots, x_n) = x_1^k + x_2^k + \cdots + x_n^k.$$

It is straightforward to show that any sum, difference or product of symmetric polynomials is a symmetric polynomial. In fact, the symmetric polynomials form a ring.

If the symmetric polynomial $\psi(\bar{x}) = \psi(x_1, x_2, \cdots, x_n)$ includes the monomial

$$a x_1^{p_1} x_2^{p_2} \cdots x_n^{p_n}$$

among its terms, then it includes also the monomial

$$a x_{\alpha_1}^{p_1} x_{\alpha_2}^{p_2} \cdots x_{\alpha_n}^{p_n}$$

where $\bar{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_n)$ is an arbitrary permutation of the integers $(1, 2, \cdots, n)$. If for a given $\bar{p} = (p_1, p_2, \cdots, p_n)$ the sum of all distinct monomials of this form is designated by

$$\phi_{\bar{p}}(\bar{x}) = \sum x_{\alpha_1}^{p_1} x_{\alpha_2}^{p_2} \cdots x_{\alpha_n}^{p_n}, \qquad \text{Eq. C.6}$$

then $\phi_{\bar{p}}(\bar{x})$ is symmetric, and further, the arbitrary symmetric polynomial $\psi(\bar{x})$ can be written as a linear combination of a finite number of such polynomials. That is

$$\psi(\bar{x}) = \sum_{\bar{p}} a_{\bar{p}} \phi_{\bar{p}}(\bar{x}).$$

A transposition of the ordered list of n variables x_k, $1 \le k \le n$ is a permutation which exchanges the positions of any two of the x_k's. Every permutation of the ordered list can be written as a composition of transpositions applied to $(1, 2, \cdots, n)$, and for any specified permutation, if any such composition includes an odd number of transpositions, then every such composition includes an odd number of transpositions. Similarly, if any such

composition includes an even number of transpositions, then every such composition includes an even number of transpositions. A permutation is designated odd or even depending upon whether its decomposition into transpositions yields an odd or even number of factors respectively. If n > 1, then exactly n!/2 odd and n!/2 even permutations exist [MoSt64].

A polynomial γ in n variables possessing the property that

$$\gamma(x_1, x_2, \cdots, x_n) = -\gamma(x_{\alpha_1}, x_{\alpha_2}, \cdots, x_{\alpha_n}) \qquad \text{Eq. C.7}$$

for every odd permutation $\bar{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_n)$ of its argument list is an alternating polynomial. It follows that any sum or difference of alternating polynomials is an alternating polynomial, that the product of a symmetric polynomial and an alternating polynomial is an alternating polynomial and that the product of any odd number of alternating polynomials is an alternating polynomial. The product of any even number of alternating polynomials is symmetric.

If the alternating polynomial $\gamma(\bar{x}) = \gamma(x_1, x_2, \cdots, x_n)$ includes the monomial

$$a x_1^{p_1} x_2^{p_2} \cdots x_n^{p_n}$$

among its terms, then it includes also the monomial

$$(-1)^{s(\bar{\alpha})} a x_{\alpha_1}^{p_1} x_{\alpha_2}^{p_2} \cdots x_{\alpha_n}^{p_n}$$

where $\bar{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_n)$ is an arbitrary permutation of the integers $(1, 2, \cdots, n)$ and where $s(\bar{\alpha})$ is the number of transpositions in the permutation $\bar{\alpha}$. If for a given $\bar{p} = (p_1, p_2, \cdots, p_n)$ the sum of all distinct monomials of this form is designated by

$$\rho_{\bar{p}}(\bar{x}) = \sum (-1)^{s(\bar{\alpha})} x_{\alpha_1}^{p_1} x_{\alpha_2}^{p_2} \cdots x_{\alpha_n}^{p_n}, \qquad \text{Eq. C.8}$$

then $\rho_{\bar{p}}(\bar{x})$ is alternating, and further, the arbitrary alternating polynomial $\gamma(\bar{x})$ can be written as a linear combination of a finite number of such polynomials. That is

$$\gamma(\bar{x}) = \sum_{\bar{p}} a_{\bar{p}} \rho_{\bar{p}}(\bar{x}).$$

As pointed out in the concluding paragraph of Section C.2, the polynomial function (defined by the product in Eq. C.4) which represents the value of an order n Vandermonde determinant alternates sign with each exchange of two variables in its argument list. Thus, it is an alternating polynomial. In fact the polynomial function which represents a Vandermonde determinant is an elementary alternating polynomial in the sense defined by the following theorem, proof of which is provided in [Aitk54] and [Muir60].

Theorem C.1: If γ is an alternating polynomial in the ordered list of n variables $x_k, 1 \le k \le n$, then γ can be written as

$$\gamma(x_1, x_2, \cdots, x_n) = \psi(x_1, x_2, \cdots, x_n) \times \prod_{k=2, j<k}^{k=n} (x_k - x_j)$$

$$= \psi(\bar{x}) \times |\bar{V}_n(\bar{x})|$$

where ψ is a symmetric polynomial.

C.4 Quasi-Symmetric (and Quasi-Alternating) Polynomials

The definitions of symmetric and alternating polynomials supplied in Section C.3 require that the relevant properties (Eq. C.5 and Eq. C.7) apply for all n! permutations of the integers $(1, 2, \cdots, n)$. This section generalizes those notions to multivariate analogs, suitable for the discussion presented in Section 9.5. The generalization amounts to restricting the applicability of Eq. C.5 and C.7 to a subset of the n! permutations of $(1, 2, \cdots, n)$. The resulting polynomial classes are referred to here as quasi-symmetric and quasi-alternating polynomials respectively.

Let ψ be a polynomial function of $n = mk$ scalar variables $x_{ij}, 1 \le i \le m, 1 \le j \le k$ and let ψ be denoted $\psi(\bar{x}_1, \bar{x}_2, \cdots, \bar{x}_m)$ where $\bar{x}_i$ is a k-component vector composed from the scalars $x_{ij}, 1 \le j \le k$. Then, ψ is quasi-symmetric if

$$\psi(\bar{x}_1, \bar{x}_2, \cdots, \bar{x}_m) = \psi(\bar{x}_{\alpha_1}, \bar{x}_{\alpha_2}, \cdots, \bar{x}_{\alpha_m}) \qquad \text{Eq. C.9}$$

for all permutations $\bar{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_m)$ of the integers $(1, 2, \cdots, m)$. The set of all m! such permutations can be placed in one-to-one correspondence with a subset of the set of all n! = (mk)! permutations of $(1, 2, \cdots, n)$.

If the quasi-symmetric polynomial $\psi(\bar{x}) = \psi(\bar{x}_1, \bar{x}_2, \cdots, \bar{x}_m)$ includes the monomial

$$a\left(x_{11}^{p_{11}} x_{12}^{p_{12}} \cdots x_{1k}^{p_{1k}}\right)\left(x_{21}^{p_{21}} x_{22}^{p_{22}} \cdots x_{2k}^{p_{2k}}\right) \cdots \left(x_{m1}^{p_{m1}} x_{m2}^{p_{m2}} \cdots x_{mk}^{p_{mk}}\right)$$

among its terms, then it includes also the monomial

$$a\left(x_{\alpha_1 1}^{p_{11}} x_{\alpha_1 2}^{p_{12}} \cdots x_{\alpha_1 k}^{p_{1k}}\right)\left(x_{\alpha_2 1}^{p_{21}} x_{\alpha_2 2}^{p_{22}} \cdots x_{\alpha_2 k}^{p_{2k}}\right) \cdots \left(x_{\alpha_m 1}^{p_{m1}} x_{\alpha_m 2}^{p_{m2}} \cdots x_{\alpha_m k}^{p_{mk}}\right)$$

where $\bar{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_m)$ is an arbitrary permutation of the integers $(1, 2, \cdots, m)$. If for a given

$$\bar{p} = (\bar{p}_1, \bar{p}_2, \cdots, \bar{p}_m) = (p_{11}, p_{12}, \cdots, p_{1k}, p_{21}, p_{22}, \cdots, p_{2k}, \cdots, p_{m1}, p_{m2}, \cdots, p_{mk})$$

the sum of all distinct monomials of this form is designated by

$$\phi_{\bar{p}}(\bar{x}) = \sum\left(x_{\alpha_1 1}^{p_{11}} x_{\alpha_1 2}^{p_{12}} \cdots x_{\alpha_1 k}^{p_{1k}}\right)\left(x_{\alpha_2 1}^{p_{21}} x_{\alpha_2 2}^{p_{22}} \cdots x_{\alpha_2 k}^{p_{2k}}\right) \cdots \left(x_{\alpha_m 1}^{p_{m1}} x_{\alpha_m 2}^{p_{m2}} \cdots x_{\alpha_m k}^{p_{mk}}\right), \qquad \text{Eq. C.10}$$

then $\phi_{\bar{p}}(\bar{x})$ is quasi-symmetric, and further, the arbitrary quasi-symmetric polynomial $\psi(\bar{x})$ can be written as a linear combination of a finite number of such polynomials. That is

$$\psi(\bar{x}) = \sum_{\bar{p}} a_{\bar{p}} \phi_{\bar{p}}(\bar{x}).$$

Eq. C.9 and C.10 are analogous to Eq. C.5 and C.6. The counterparts of Eq. C.7 and Eq. C.8 can be developed by straightforward extension of these results. Thus, let γ be a polynomial function of n = mk scalar variables x_{ij}, $1 \le i \le m$, $1 \le j \le k$ and let γ be denoted $\gamma(\bar{x}_1, \bar{x}_2, \cdots, \bar{x}_m)$ where $\bar{x}_i$ is a k-component vector composed from the scalars x_{ij}, $1 \le j \le k$. Then, γ is quasi-alternating if

$$\gamma(\bar{x}_1, \bar{x}_2, \cdots, \bar{x}_m) = -\gamma(\bar{x}_{\alpha_1}, \bar{x}_{\alpha_2}, \cdots, \bar{x}_{\alpha_m}) \qquad \text{Eq. C.11}$$

for every odd permutation $\bar{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_m)$ of the integers $(1, 2, \cdots, m)$. The set of all m!/2 such permutations can be placed in one-to-one correspondence with a subset of the set of all n!/2 = (mk)!/2 odd permutations of $(1, 2, \cdots, n)$.

If the quasi-alternating polynomial $\gamma(\bar{x}) = \gamma(\bar{x}_1, \bar{x}_2, \cdots, \bar{x}_m)$ includes the monomial

$$a\left(x_{11}^{p_{11}} x_{12}^{p_{12}} \cdots x_{1k}^{p_{1k}}\right)\left(x_{21}^{p_{21}} x_{22}^{p_{22}} \cdots x_{2k}^{p_{2k}}\right) \cdots \left(x_{m1}^{p_{m1}} x_{m2}^{p_{m2}} \cdots x_{mk}^{p_{mk}}\right)$$

among its terms, then it includes also the monomial

$$(-1)^{s(\bar{\alpha})} a\left(x_{\alpha_1 1}^{p_{11}} x_{\alpha_1 2}^{p_{12}} \cdots x_{\alpha_1 k}^{p_{1k}}\right)\left(x_{\alpha_2 1}^{p_{21}} x_{\alpha_2 2}^{p_{22}} \cdots x_{\alpha_2 k}^{p_{2k}}\right) \cdots \left(x_{\alpha_m 1}^{p_{m1}} x_{\alpha_m 2}^{p_{m2}} \cdots x_{\alpha_m k}^{p_{mk}}\right)$$

where $\bar{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_m)$ is an arbitrary permutation of the integers $(1, 2, \cdots, m)$ and where $s(\bar{\alpha})$ is the number of transpositions in the permutation $\bar{\alpha}$. If for a given

$$\bar{p} = (\bar{p}_1, \bar{p}_2, \cdots, \bar{p}_m) = (p_{11}, p_{12}, \cdots, p_{1k}, p_{21}, p_{22}, \cdots, p_{2k}, \cdots, p_{m1}, p_{m2}, \cdots, p_{mk})$$

the sum of all <u>distinct</u> monomials of this form is designated by

$$\rho_{\bar{p}}(\bar{x}) = \sum (-1)^{s(\bar{\alpha})} \left(x_{\alpha_1 1}^{p_{11}} x_{\alpha_1 2}^{p_{12}} \cdots x_{\alpha_1 k}^{p_{1k}}\right)\left(x_{\alpha_2 1}^{p_{21}} x_{\alpha_2 2}^{p_{22}} \cdots x_{\alpha_2 k}^{p_{2k}}\right) \cdots \left(x_{\alpha_m 1}^{p_{m1}} x_{\alpha_m 2}^{p_{m2}} \cdots x_{\alpha_m k}^{p_{mk}}\right), \qquad \text{Eq. C.12}$$

then $\rho_{\bar{p}}(\bar{x})$ is quasi-alternating, and further, the arbitrary quasi-alternating polynomial $\gamma(\bar{x})$ can be written as a linear combination of a finite number of such polynomials. That is

$$\gamma(\bar{x}) = \sum_{\bar{p}} a_{\bar{p}} \rho_{\bar{p}}(\bar{x}).$$

APPENDIX D

COMPUTER LISTINGS

D.1 Introduction

This appendix includes listings of the computer programs used to generate the simulation data presented in Section 5. These programs were developed on the Eglin AFB, Fl. Cray Y-MP. The programs are all written in Fortran and in some cases they employ Cray extensions to the Fortran standard. The listings are separated into two subsections, one including main program listings and a second including the contents of a subprogram library accessed by the main programs. The library procedures section also includes a library table of contents.

D.2 Main Program Listings

```
      PROGRAM GET_NPS
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     Name: GET_NPS
C
C     Purpose: Compute (via binomial coefficient) and output
C              the cardinality of the indicated S's and S''s
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     Declare local variables
C
      INTEGER M, L, NN, NP, NPP
      DOUBLE      FNP, FNPP
C
C     Loop over M = 1 to M = 8
C
      DO 2 M = 1, 8
      WRITE( 6, * ) 'M = ', M
C
C     Get the answers and write them to stdout
C
      DO 1 L = 1, 8
      NN = 2**L + M
      FNP = 1.0
      FNPP = 1.0
```

```
      DO 4 J = 0, M - 1
      FNP = FNP*FLOAT( NN - 1 - J )/FLOAT( M - J )
 4    FNPP = FNPP*FLOAT( NN - J )/FLOAT( M - J )
      NP = 0.5 + FNP
      NPP = 0.5 + FNPP
 1    WRITE( 6, 3 ) L, 2**L, NP, NPP
 2    WRITE( 6, * )
 3    FORMAT( 5H L = , I4, 5H N = , I8,
     &        6H NP = , I22, 7H NPP = , I22 )
C
C     Finished
C
      END

      PROGRAM GET_SPS
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     Name: GET_SPS
C
C     Purpose: Compute and output the indicated S's
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     Declare local variables
C
      INTEGER M, L, NP, LM, LL, LN, J
      INTEGER SP( 4*5984 ), NBAR( 16 )
C
C     Prompt for and read ranges
C
      WRITE( 6, * ) ' Maximum M? '
      READ( 5, '(I8)' ) M
      WRITE( 6, * ) ' Maximum L? '
      READ( 5, '(I8)' ) L
C
C     Get the answers and write them to stdout
C
      DO 3 LM = 1, M
      DO 2 LL = 1, L
      CALL GET_SP( LM, LL, SP, NP )
      WRITE( 6, * )
      WRITE( 6, 4 ) LM, LL, 2**LL, NP
      DO 1 LN = 1, NP
      CALL UNPACK_NBARP( LM, LL, SP( ( LN - 1 )*LM + 1 ), NBAR )
 1    WRITE( 6, 5 ) ( NBAR( J ), J = 1, 2**LL )
 2    CONTINUE
 3    CONTINUE
 4    FORMAT( 5H M = , I4, 5H L = , I4, 5H N = , I4, 6H NP = , I12 )
 5    FORMAT( 16I4 )
C
C     Finished
C
      END
```

```
      PROGRAM GET_R
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     Name: GET_R
C
C     Purpose: Generate the indicated reward function and write
C              it to disk
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     Declare local variables
C
      INTEGER L, SCHEMA, SCHEMA_MASK, DELTA
      REAL R( 0:2**8 - 1 ), WEIGHT
      CHARACTER*8 SCHEMAC
C
C     Prompt for and read the bit-string length
C
      WRITE( 6, * ) 'Bit-string length? '
      READ( 5, * ) L
C
C     Loop over all schemata and retrieve the associated weight
C
      DO 4 K = 1, 3**L
      DO 1 N = 1, L
      INDICATOR = MOD( ( K - 1 )/3**( N - 1 ), 3 )
      IF( INDICATOR .EQ. 0 ) THEN
            SCHEMAC( N:N ) = '*'
      ELSE IF( INDICATOR .EQ. 1 ) THEN
            SCHEMAC( N:N ) = '0'
      ELSE
            SCHEMAC( N:N ) = '1'
      END IF
1     CONTINUE
      WRITE( 6, * ) SCHEMAC( 1:N )
      READ( 5, * ) WEIGHT
C
C     Build the schema and schema mask
C
      SCHEMA = 0
      SCHEMA_MASK = 0
      DO 3 N = 1, L
      DELTA = 2**( N - 1 )
      IF( SCHEMAC( N:N ) .EQ. '1' ) THEN
            SCHEMA = SCHEMA + DELTA
            SCHEMA_MASK = SCHEMA_MASK + DELTA
      ELSE IF( SCHEMAC( N:N ) .EQ. '0' ) THEN
            SCHEMA_MASK = SCHEMA_MASK + DELTA
      END IF
3     CONTINUE
C
C     Now add required contributions to R
C
      DO 4 I = 0, 2**L - 1
      IF( AND( SCHEMA_MASK, XOR( I, SCHEMA ) ) .EQ. 0 )
```

```
     &          R( I ) = R( I ) + WEIGHT
4    CONTINUE
5    CONTINUE
C
C    Open the output data file and write R
C
     OPEN( 1, FILE='RDATA', STATUS='NEW' )
     WRITE( 1, '(8F10.6)' ) ( R( J ), J = 0, 2**L - 1 )
C
C    Finished
C
     END

     PROGRAM GET_P2INS
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C    Name: GET_P2INS
C
C    Purpose: Compute and return the indicated conditional
C             probability arrays
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C    Declare problem defining parameters
C
     INTEGER M, L, NP, NALPHA
     PARAMETER ( M = 6, L = 4, NP = 54264, NALPHA = 10 )
     INTEGER SP( M, NP )
     REAL R( 0:2**L - 1 ), P2IN( 0:2**L - 1 )
     INTEGER NCOUNTS( 4 )
     DATA NCOUNTS/1,3096,3100,54181/
     CHARACTER*8 P2INDATA
     DATA P2INDATA/'P2INDATA'/
C
C    Declare local variables
C
     INTEGER NBAR( 0:15 )
     REAL ALPHA
C
C    Get the objective function values
C
     OPEN( 1, FILE='RDATA', STATUS='OLD' )
     READ( 1, '(8F10.6)' ) ( R( I ), I = 0, 2**L - 1 )
     CLOSE( 1 )
C
C    Open the output data file and write the summary data
C
     OPEN( 1, FILE=P2INDATA, STATUS='NEW' )
     WRITE( 1, '(4I8)' ) M, L, NP, NALPHA
     WRITE( 1, '(8F10.6)' ) R
     WRITE( 1, * )
C
C    Build the state space array
C
     CALL GET_SP( M, L, SP )
```

```
C
C       Generate the required P2IN data
C
        DO 2 NCOUNT = 1, 4
        CALL UNPACK_NBARP( M, L, SP( 1, NCOUNTS( NCOUNT ) ), NBAR )
        WRITE( 1, '(16I2)' ) NBAR
        DO 1 I = 0, NALPHA - 1
        ALPHA = 1.0/2**I
        WRITE( 1, '(F10.6)' ) ALPHA
        CALL GET_P2IN( M, L, SP( 1, NCOUNTS( NCOUNT ) ),
     &             ALPHA, R, P2IN )
        WRITE( 1, '(8F10.6)' ) P2IN
1       WRITE( 1, * )
2       CONTINUE
C
C       Finished
C
        END

        PROGRAM GET_P3INS
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       Name: GET_P3INS
C
C       Purpose: Compute and return the indicated conditional
C                probability arrays
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       Declare problem defining parameters
C
        INTEGER M, L, NP, NALPHA
        PARAMETER ( M = 6, L = 4, NP = 54264, NALPHA = 10 )
        INTEGER SP( M, NP )
        REAL R( 0:2**L - 1 ), P3IN( 0:2**L - 1 )
        INTEGER NCOUNTS( 4 )
        DATA NCOUNTS/1,3096,3100,54181/
        CHARACTER*8 P3INDATA
        DATA P3INDATA/'P3INDATA'/
C
C       Declare local variables
C
        INTEGER NBAR( 0:15 )
        REAL ALPHA
C
C       Get the objective function values
C
        OPEN( 1, FILE='RDATA', STATUS='OLD' )
        READ( 1, '(8F10.6)' ) ( R( I ), I = 0, 2**L - 1 )
        CLOSE( 1 )
C
C       Open the output data file and write the summary data
C
        OPEN( 1, FILE=P3INDATA, STATUS='NEW' )
        WRITE( 1, '(4I8)' ) M, L, NP, NALPHA
```

```
      WRITE( 1, '(8F10.6)' ) R
      WRITE( 1, * )
C
C     Build the state space array
C
      CALL GET_SP( M, L, SP )
C
C     Generate the required P3IN data
C
      DO 2 NCOUNT = 1, 4
      CALL UNPACK_NBARP( M, L, SP( 1, NCOUNTS( NCOUNT ) ), NBAR )
      WRITE( 1, '(16I2)' ) NBAR
      DO 1 I = 0, NALPHA - 1
      ALPHA = 1.0/2**I
      WRITE( 1, '(F10.6)' ) ALPHA
      CALL GET_P3IN( M, L, SP( 1, NCOUNTS( NCOUNT ) ),
     &           ALPHA, R, P3IN )
      WRITE( 1, '(8F10.6)' ) P3IN
1     WRITE( 1, * )
2     CONTINUE
C
C     Finished
C
      END

      PROGRAM GET_3STAT
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     Name: GET_3STAT
C
C     Purpose: Compute the indicated three-operator stationary
C                    distribution
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     Declare defining parameters and output file name
C
      INTEGER M, L, NP
      PARAMETER ( M = 4, L = 5, NP = 52360 )
      REAL ALPHA, ALPHA0, R( 0:2**L - 1 ), TDELTA
      INTEGER NALPHA
      PARAMETER ( ALPHA0 = 1.0/2**20, NALPHA = 1, TDELTA = 0.004 )
      CHARACTER*10 TESTFILE
      DATA TESTFILE/'3TEST45'/
C
C     Declare state space associated arrays
C
      INTEGER SP( M, NP ), SPA( 0:2**L - 1 ), MULTI( NP )
      REAL QBAR( NP, 0:1 )
C
C     Declare some local variables
C
      REAL LDELTA
      INTEGER TOGGLE, LOOP_COUNT
C
```

```
C       Get the objective function data
C
        OPEN( 1, FILE='RDATA4', STATUS='OLD' )
        READ( 1, '(8F10.6)' ) ( R( J ), J = 0, 2**L - 1 )
        CLOSE( 1 )
C
C       Open the output data file and write the summary output data
C
        OPEN( 1, FILE=TESTFILE, STATUS = 'NEW', FORM='UNFORMATTED' )
        WRITE( 1 ) M, L, NP
        WRITE( 1 ) ALPHA0, NALPHA, TDELTA
        WRITE( 1 ) R
C
C       Generate and store the state space set, S'
C
        CALL GET_SP( M, L, SP )
C
C       Next, get the indices of the absorbing states in S'
C
        CALL GET_SPA( M, L, NP, SP, SPA )
C
C       Now get the associated multinomial coefficient array
C
        CALL GET_MULTI( M, L, NP, SP, MULTI )
C
C       Compute for each required ALPHA
C
        DO 3 K = 0, NALPHA - 1
C
C       Initialize ALPHA and QBAR
C
        ALPHA = ALPHA0/2**K
        CALL INIT_QBAR( M, L, NP, MULTI, QBAR )
C
C       Loop until the tolerance parameter is met
C
        TOGGLE = 0
        LOOP_COUNT = 0
        LDELTA = 1.0
        DO 1 WHILE ( LDELTA .GT. TDELTA )
        CALL GET_3QBAR( M, L, NP, ALPHA, R, MULTI,
     &              SP, SPA, QBAR( 1, TOGGLE ),
     &              QBAR( 1, MOD( TOGGLE + 1, 2 ) ),
     &              LDELTA )
        TOGGLE = MOD( TOGGLE + 1, 2 )
        LOOP_COUNT = LOOP_COUNT + 1
        WRITE(6,*) LOOP_COUNT, TDELTA, LDELTA
1       CONTINUE
C
C       Output the termination information and the final vector
C
        WRITE( 1 ) ALPHA, LOOP_COUNT - 1, LDELTA
        DO 2 I = 1, NP/40
2       WRITE( 1 ) ( QBAR( 40*( I - 1 ) + J, TOGGLE ), J = 1, 40 )
        WRITE( 1 ) ( QBAR( J, TOGGLE ), J = 40*( NP/40 ) + 1, NP )
```

```
3       CONTINUE
C
C       Last write the absorbing state vector values
C
        WRITE( 1 ) ( QBAR( SPA( I ), TOGGLE ), I = 0, 2**L - 1 )
C
C       Finished
C
        END
```

D.3 Library Listings

```
stat.o:GET_SP
stat.o:GET_SPA
stat.o:GET_MULTI
stat.o:INIT_QBAR
stat.o:GET_3QBAR
stat.o:INIT_NBARP
stat.o:GET_NBARP
stat.o:GET_NFAC
stat.o:GET_P3MN
stat.o:GET_P3IN
stat.o:GET_P1IN
stat.o:UNPACK_NBARP

        SUBROUTINE GET_SP( M, L, SP, NP )
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       Name: GET_SP
C
C       Purpose: Generate S'
C
C       Note: The fourth argument (NP) is optional
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       Declare calling arguments
C
        INTEGER M, L, SP( M, 0:* ), NP
C
C       Declare local variables
C
        INTEGER NCOUNT, NBARP( M ), I, NARG
        LOGICAL LSTAT
C
C       Initialize NBARP
C
        CALL INIT_NBARP( M, NBARP )
C
C       Loop until S' is complete
C
        NCOUNT = 0
        LSTAT = .TRUE.
        DO 2 WHILE( LSTAT )
C
```

```
C       Set this element in S'
C
        DO 1 I = 1, M
1       SP( I, NCOUNT ) = NBARP( I )
C
C       Get the next one and increment the counter
C
        CALL GET_NBARP( M, L, NBARP, LSTAT )
2       NCOUNT = NCOUNT + 1
C
C       Test argument count to determine whether to set NP
C
        NARG = NUMARG()
        IF( NARG .EQ. 4 ) NP = NCOUNT
C
C       Return to caller
C
        RETURN
        END

        SUBROUTINE GET_SPA( M, L, NP, SP, SPA )
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       Name: GET_SPA
C
C       Purpose: Generate a table of absorbing state indices in S'
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       Declare calling arguments
C
        INTEGER M, L, NP, SP( M, NP ), SPA( 0:2**L - 1 )
C
C       Declare some local variables
C
        INTEGER I, J, K, JSTART
C
C       Initialize
C
        JSTART = 1
C
C       Loop over all I in S
C
        DO 3 I = 1, 2**L
C
C       Loop over all J in S'
C
        DO 2 J = JSTART, NP
C
C       Test SP until an exit condition is satisfied
C
        DO 1 K = 1, M
        IF( SP( K, J ) .NE. I ) GO TO 2
1       CONTINUE
C
```

```
C       Exhausting M signals an absorbing state
C       Assign it and go after the next one
C
        SPA( I - 1 ) = J
        JSTART = J + 1
        GO TO 3
C
C       Exit to label 2 means that this J is not an
C       absorbing state
C
2       CONTINUE
3       CONTINUE
C
C       Return to caller
C
        RETURN
        END

        SUBROUTINE GET_MULTI( M, L, NP, SP, MULTI )
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       Name: GET_MULTI
C
C       Purpose: Get the multinomial coefficient table for the
C                        supplied S'
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       Declare calling arguments
C
        INTEGER M, L, NP, SP( M, NP ), MULTI( NP )
C
C       Declare local variables and function reference
C
        INTEGER MFAC, NBAR( 0:2**L - 1 ), NCOUNT
        INTEGER I, GET_NFAC
C
C       Compute and save M!
C
        MFAC = GET_NFAC( M )
C
C       Loop over all vectors in S'
C
        DO 2 NCOUNT = 1, NP
C
C       Set MULTI( NCOUNT ) = M!
C
        MULTI( NCOUNT ) = MFAC
C
C       Now unpack this NBARP
C
        CALL UNPACK_NBARP( M, L, SP( 1, NCOUNT ), NBAR )
C
C       Loop over the denominator factorials
C
```

```
      DO 1 I = 0, 2**L - 1
1     MULTI( NCOUNT ) = MULTI( NCOUNT )/GET_NFAC( NBAR( I ) )
C
C     Close the loop
C
2     CONTINUE
C
C     Return to caller
C
      RETURN
      END

      SUBROUTINE INIT_QBAR( M, L, NP, MULTI, QBAR )
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     Name: INIT_QBAR
C
C     Purpose: Initialize the QBAR probability vectors
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     Declare calling arguments
C
      INTEGER M, L, NP, MULTI( NP )
      REAL QBAR( NP, 0:1 )
C
C     Declare a local variable
C
      REAL FRACTION
C
C     Set QBAR1 to its ALPHA = 1 value and zero QBAR2
C
      FRACTION = 1.0/( 2**( M*L ) )
      DO 1 I = 1, NP
      QBAR( I, 0 ) = MULTI( I )*FRACTION
1     QBAR( I, 1 ) = 0.0
C
C     Return to caller
C
      RETURN
      END

      SUBROUTINE GET_3QBAR( M, L, NP, ALPHA, R, MULTI,
     &                      SP, SPA, QBAR0, QBAR1, LDELTA )
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     Name: GET_3QBAR
C
C     Purpose: Compute and return the indicated three-operator
C              stationary distribution vector transformation
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     Declare calling arguments
C
```

```
      INTEGER M, L, NP, MULTI( NP ), SP( M, NP ), SPA( 0:2**L - 1 )
      REAL ALPHA, R( 0:2**L - 1 )
      REAL QBAR0( NP ), QBAR1( NP ), LDELTA
C
C     Declare local variables
C
      INTEGER MCOUNT, NCOUNT
      REAL SIGMA, P3MN( NP )
C
C     Clear QBAR1
C
      DO 1 MCOUNT = 1, NP
1     QBAR1( MCOUNT ) = 0.0
C
C     Loop over all states in S'
C
      DO 3 NCOUNT = 1, NP
C
C     Get P3MN for this NBAR
C
      CALL GET_P3MN( M, L, NP, ALPHA, R, MULTI, NCOUNT, SP, P3MN )
C
C     Accumulate them
C
      DO 2 MCOUNT = 1, NP
2     QBAR1( MCOUNT ) = QBAR1( MCOUNT ) +
     &          QBAR0( NCOUNT ) * P3MN( MCOUNT )
C
C     Close the NBAR loop
C
3     CONTINUE
C
C     Normalize
C
      SIGMA = 0.0
      DO 4 MCOUNT = 1, NP
4     SIGMA = SIGMA + QBAR1( MCOUNT )
      DO 5 MCOUNT = 1, NP
5     QBAR1( MCOUNT ) = QBAR1( MCOUNT )/SIGMA
C
C     Reset LDELTA
C
      SIGMA = 0.0
      DO 6 I = 0, 2**L - 1
6     SIGMA = SIGMA + QBAR1( SPA( I ) )
      LDELTA = 1.0 - SIGMA
C
C     Return to caller
C
      RETURN
      END
```

```
      SUBROUTINE INIT_NBARP( M, NBARP )
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     Name: INIT_NBARP
C
C     Purpose: Initialize NBARP to the starting element of S'
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     Declare calling arguments
C
      INTEGER M, NBARP( M )
C
C     Set all pointers in the NBARP array to cell 1
C
      DO 1 I = 1, M
1     NBARP( I ) = 1
C
C     Return to caller
C
      RETURN
      END

      SUBROUTINE GET_NBARP( M, L, NBARP, NSTAT )
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     Name: GET_NBARP
C
C     Purpose: Generate the successor element in S'
C
C     Note:   (1) This procedure assumes that M and L are in range
C                     and that the caller supplied NBARP is valid
C
C             (2) The fourth argument (NSTAT) is optional
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     Declare calling arguments
C
      INTEGER M, L, NBARP( M )
      LOGICAL NSTAT
C
C     Declare local variables
C
      INTEGER IMAX, I, J, NARG
      LOGICAL LSTAT
C
C     Set maximum index
C
      IMAX = 2**L
C
C     Process most frequent transition
C
      IF( NBARP( M ) .LT. IMAX ) THEN
          NBARP( M ) = NBARP( M ) + 1
```

```
            LSTAT = .TRUE.
C
C       Next most frequent
C
        ELSE IF( NBARP( 1 ) .LT. IMAX ) THEN
            DO 1 I = M - 1, 1, -1
            IF( NBARP( I ) .LT. IMAX ) GO TO 2
1           CONTINUE
2           NBARP( I ) = NBARP( I ) + 1
            DO 3 J = I + 1, M
3           NBARP( J ) = NBARP( I )
            LSTAT = .TRUE.
C
C       Anything else is terminal
C
        ELSE
            LSTAT = .FALSE.
        END IF
C
C       Test argument count to determine whether to set NSTAT
C
        NARG = NUMARG()
        IF( NARG .EQ. 4 ) NSTAT = LSTAT
C
C       Return to caller
C
        RETURN
        END

        INTEGER FUNCTION GET_NFAC( N )
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       Name: GET_NFAC
C
C       Purpose: Compute and return N!
C
C       Note: This procedure assumes that N is a nonnegative integer
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       Declare calling argument
C
        INTEGER N
C
C       If N = 0 or N = 1, then N! = 1
C
        IF( N .LE. 1 ) THEN
            GET_NFAC = 1
C
C       Otherwise, recurse
C
        ELSE
            GET_NFAC = N * GET_NFAC( N - 1 )
        END IF
C
```

```
C       Return to caller
C
        RETURN
        END

        SUBROUTINE GET_P3MN( M, L, NP, ALPHA, R, MULTI,
     &                       NCOUNT, SP, P3MN )
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       Name: GET_P3MN
C
C       Purpose: Compute and return the indicated conditional
C                probability array
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       Declare calling arguments
C
        INTEGER M, L, NP, MULTI( NP ), NCOUNT, SP( M, NP )
        REAL ALPHA, R( 0:2**L - 1 ), P3IN( 0:2**L - 1 ), P3MN( NP )
C
C       Declare local variables
C
        INTEGER MCOUNT, K
C
C       Get P3IN for this NBARP
C
        CALL GET_P3IN( M, L, SP( 1, NCOUNT ), ALPHA, R, P3IN )
C
C       Initialize the P3MN vector
C
        DO 1 MCOUNT = 1, NP
1       P3MN( MCOUNT ) = MULTI( MCOUNT )
C
C       Loop over the solutions represented in this NBARP
C
        DO 3 J = 1, M
        DO 2 MCOUNT = 1, NP
        K = SP( J, MCOUNT ) - 1
2       P3MN( MCOUNT ) = P3MN( MCOUNT )*P3IN( K )
3       CONTINUE
C
C       Return to caller
C
        RETURN
        END

        SUBROUTINE GET_P3IN( M, L, NBARP, ALPHA, R, P3IN )
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       Name: GET_P3IN
C
C       Purpose: Compute and return the indicated conditional
C                probability array
C
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       Declare calling arguments
C
        INTEGER M, L, NBARP( M )
        REAL ALPHA, R( 0:2**L - 1 ), P3IN( 0:2**L - 1 )
C
C       Declare local variables
C
        INTEGER I, J, K, IMAX, INDX
        REAL P1IN( 0:2**L - 1 ), P3INP( 0:2**L - 1 )
        REAL P2, SUM
C
C       Set maximum loop index
C
        IMAX = 2**L - 1
C
C       Clear the P3IN and P3INP arrays
C
        DO 1 I = 0, IMAX
        P3IN( I ) = 0.0
1       P3INP( I ) = 0.0
C
C       Get the P1IN array
C
        CALL GET_P1IN( M, L, NBARP, R, P1IN )
C
C       Build the P3INP array
C
        DO 4 I = 0, IMAX
        DO 3 J = 0, IMAX
        P2 = P1IN( I )*P1IN( J )
        DO 2 K = 0, L
        INDX = I
        CALL MVBITS( J, 0, K, INDX, 0 )
2       P3INP( INDX ) = P3INP( INDX ) + P2
3       CONTINUE
4       CONTINUE
C
C       Perturb with mutation
C
        DO 6 J = 0, IMAX
        IF( P3INP( J ) .NE. 0.0 ) THEN
                DO 5 I = 0, IMAX
5               P3IN( I ) = P3IN( I ) + P3INP( J )*
     &                  ALPHA**POPCNT( XOR( I, J ) )
        END IF
6       CONTINUE
C
C       Now normalize them
C
        SUM = 0.0
        DO 7 I = 0, IMAX
7       SUM = SUM + P3IN( I )
        DO 8 I = 0, IMAX
```

```
8       P3IN( I ) = P3IN( I )/SUM
C
C       Return to caller
C
        RETURN
        END

        SUBROUTINE GET_P1IN( M, L, NBARP, R, P1IN )
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       Name: GET_P1IN
C
C       Purpose: Compute and return the indicated conditional
C                probability array
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       Declare calling arguments
C
        INTEGER M, L, NBARP( M )
        REAL R( 0:2**L - 1 ), P1IN( 0:2**L - 1 )
C
C       Declare local variables
C
        INTEGER I, J, K, IMAX
        REAL SUM
C
C       Set maximum loop index
C
        IMAX = 2**L - 1
C
C       Clear the P1IN vector
C
        DO 1 I = 0, IMAX
1       P1IN( I ) = 0.0
C
C       Compute the numerators and accumulate the denominator
C
        SUM = 0.0
        DO 2 J = 1, M
        K = NBARP( J ) - 1
        P1IN( K ) = P1IN( K ) + R( K )
2       SUM = SUM + R( K )
C
C       Now normalize them
C
        DO 3 I = 0, IMAX
3       P1IN( I ) = P1IN( I )/SUM
C
C       Return to caller
C
        RETURN
        END
```

```
      SUBROUTINE UNPACK_NBARP( M, L, NBARP, NBAR )
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     Name: UNPACK_NBARP
C
C     Purpose: Generate displayable version of packed NBAR
C
C     Note:  This procedure assumes that M and L are in range
C            that the caller supplied NBARP is valid
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     Declare calling arguments
C
      INTEGER M, L, NBARP( M ), NBAR( 0:2**L - 1 )
C
C     Declare local variables
C
      INTEGER I
C
C     Clear the NBAR vector
C
      DO 1 I = 0, 2**L - 1
1     NBAR( I ) = 0
C
C     Now set the nonzero components
C
      DO 2 I = 1, M
2     NBAR( NBARP( I ) - 1 ) = NBAR( NBARP( I ) - 1 ) + 1
C
C     Return to caller
C
      RETURN
      END
```

REFERENCES

[Aitk54] Aitken, A. C., "Determinants and Matrices," Interscience Publishers, Inc., New York, N. Y., 1954.

[Beth80] Bethke, A. D., "Genetic Algorithms as Function Optimizers," Ph. D. Dissertation, University of Michigan, Ann Arbor, Mich., 1980.

[BrGo87] Bridges, C. L. and Goldberg, D. E., "An Analysis of Reproduction and Crossover in a Binary Coded Genetic Algorithm," in Grefenstette, J. J. (Ed.) "Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms", Lawrence Earlbaum Associates, Publishers, Hillsdale, N. J., 1987, pp 9-13.

[BrGo89] Bridges, C. L. and Goldberg, D. E., "A Note on the Non-Uniform Walsh Schema Transform," TCGA Report No. 89004, Jun 1989, Dept. of Engineering Mechanics, The University of Alabama, Tuscaloosa, Ala.

[Cinl75] Cinlar, E., "Introduction to Stochastic Processes," Prentice Hall, Englewood Cliffs, N. J., 1975.

[Davi87] Davis, L. (Ed.), "Genetic Algorithms and Simulated Annealing," Morgan Kaufman Publishers, Inc., Los Altos, Calif., 1987.

[Dejo75] De Jong, K. A., "An Analysis of the Behavior of a Class of Genetic Adaptive Systems," Ph. D. Dissertation, University of Michigan, Ann Arbor, Mich., 1975.

[GeGe84] Geman, S. and Geman, D., "Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images," IEEE Trans. Patt. Anal. Mach. Intel., Vol. PAMI-6, No. 6, Nov. 1984, pp 721-741.

[Gold83] Goldberg, D. E., "Computer-Aided Gas Pipeline Operation Using Genetic Algorithms and Rule Learning," Ph.D. Dissertation, University of Michigan, Ann Arbor, Mich., 1983.

[Gold85] Goldberg, D. E., "Optimal Initial Population Size for Binary Coded Genetic Algorithms," TCGA Report No 85001, Nov 1985, Dept. of Engineering Mechanics, The University of Alabama, Tuscaloosa, Ala.

[Gold87] Goldberg, D. E., "Simple Genetic Algorithms and the Minimal Deceptive Problem," in Davis, L. (Ed.), "Genetic Algorithms and Simulated Annealing," Morgan Kaufman Publishers, Inc., Los Altos, Calif., 1987.

[Gold88] Goldberg, D. E., "Genetic Algorithms and Walsh Functions: Part I, A Gentle Introduction," TCGA Report No. 88006, Nov 1988, Dept. of Engineering Mechanics, The University of Alabama, Tuscaloosa, Ala.

[Gold89a] Goldberg, D., "Genetic Algorithms in Search, Optimization and Machine Learning," Addison-Wesley Publishing Company, Inc., Reading, Mass., 1989.

[Gold89b] Goldberg, D. E., "Genetic Algorithms and Walsh Functions: Part II, Deception and its Analysis," TCGA Report No. 89001, Jan 1989, Dept. of Engineering Mechanics, The University of Alabama, Tuscaloosa, Ala.

[GoSe87] Goldberg, D. E. and Segrest, P., "Finite Markov Chain Analysis of Genetic Algorithms," in Grefenstette, J. J. (Ed.) "Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms," Lawrence Earlbaum Associates, Publishers, Hillsdale, N. J., 1987, pp 1-8.

[Gref85] Grefenstette, J. J. (Ed.), "Proceedings of an International Conference on genetic Algorithms and Their Applications," Lawrence Earlbaum Associates, Publishers, Hillsdale, N. J., 1985.

[Gref87] Grefenstette, J. J. (Ed.), "Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on genetic Algorithms," Lawrence Earlbaum Associates, Publishers, Hillsdale, N. J., 1987.

[Hall67] Hall, M., "Combinatorial Theory," Blaisdell Publishing Company, Waltham, Mass., 1967.

[Holl75] Holland, J. H., "Adaptation in Natural and Artificial Systems," University of Michigan Press, Ann Arbor, Mich., 1975.

[IsMa76] Isaacson, D. L. and Madsen, R. W., "Markov Chains Theory and Applications," John Wiley & Sons, New York, N. Y., 1976.

[KiGe83] Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P., "Optimization by Simulated Annealing," Science, Vol 220, Number 4598, May 1983, pp 671-680.

[LaAa87] Laarhoven, P. J. M. and Aarts, E. H. L., "Simulated Annealing," D. Reidel Publishing Company, Dordrecht, Holland, 1987.

[LuMe86] Lundy, M. and Mees, A., "Convergence of an Annealing Algorithm," Math. Prog., 34, 1986, pp111-124.

[Metr53] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E., "Equation of State Calculations by Fast Computing Machines," The Journal of Chemical Physics, Vol 21, Number 6, June 1953, pp 1087-1092.

[MiRo85] Mitra, D., Romeo, F. and Sangiovanni-Vincentelli, A., "Convergence and Finite Time Behavior of Simulated Annealing," Proc 24 th Conference on Decision and Control, Ft. Lauderdale, 1985, pp 761-767.

[MoSt64] Mostowski, A. and Stark, M., "Introduction to Higher Algebra," The Macmillian Company, New York, N. Y., 1964.

[Muir60] Muir, T., "A Treatise on the Theory of Determinants," Dover Publications, Inc., New York, N. Y., 1960.

[Rior58] Riordan, J., "An Introduction to Combinatorial Analysis," John Wiley & Sons, New York, N. Y., 1958.

[RoSa85] Romeo, F. and Sangiovanni-Vincentelli, A., "Probabilistic Hill Climbing Algorithms: Properties and Applications," Proc. 1985 Chapel Hill Conference on VLSI, May 1985, pp393-417.

[Sene81] Seneta, E., "Non-negative Matrices and Markov Chains," Springer Verlag, New York, N. Y., 1981.

BIOGRAPHICAL SKETCH

Tom Davis enrolled in the Purdue University School of Electrical Engineering in September, 1967 and was awarded the Bachelor of Science in Electrical Engineering degree in June, 1971. Upon graduation, he was commissioned in the United States Navy and entered flight training at Pensacola, Florida. He was designated a naval aviator in April, 1973. He subsequently completed several aviation related tours of duty, including two years as a primary flight instructor at Saufley Field, Florida, and three years in an air anti-submarine warfare squadron home-ported in Brunswick, Maine. During the later tour, he completed three extended overseas deployments to European and North Atlantic island operational sites.

Tom resigned from the navy in August, 1978 and entered the Purdue University School of Electrical Engineering as a graduate student. He was awarded the Master of Science in Electrical Engineering degree in December, 1979. In January, 1980, he reported to work at the Guided Weapons Division of the Air Force Armament Test Laboratory at Eglin AFB, Florida. During the following eight years, he was assigned to a variety of millimeter wave radar and infrared seeker development programs for autonomous targeting tactical weapons systems.

In May, 1988, Tom was admitted to the Graduate School of the University of Florida and he began attending classes in Gainesville in August of that year. During the calendar year ending in August, 1989, he satisfied the residency, course work and entrance examination for admission to the PhD program. At that time, he returned to Eglin AFB and resumed his duties in the Armament Laboratory. He was admitted to candidacy following a qualifying exam conducted at Gainesville in October, 1989, and since then has been engaged in dissertation research in stochastic relaxation search algorithms.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Jose C. Principe, Chairman
Associate Professor of Electrical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Donald G. Childers
Professor of Electrical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Antonio Arroyo
Associate Professor of Electrical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Murali Rao
Professor of Mathematics

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Eugene R. Chenette
Professor of Electrical Engineering

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

May 1991

Herbert A. Bevis
for Winfred M. Phillips
Dean, College of Engineering

Madelyn M. Lockhart
Dean, Graduate School